The Dissertation Committee for Farhad Tarahhom
certifies that this is the approved version of the following dissertation:

Development of an Implicit Full-Tensor Dual Porosity Compositional Reservoir Simulator

Committee:

Kamy Sepehrnoori, Supervisor

Gary A. Pope

Steven L. Bryant

Todd J. Arbogast

Mojdeh Delshad

Development of an Implicit Full-Tensor Dual Porosity Compositional Reservoir Simulator

by

Farhad Tarahhom, B.S., M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December 2008

To my wife, Ghazal Dashti, my father, Yadollah Tarahhom, my mother, Kobra Bahmee, and my brothers and sisters for their love, patience, and support.
To the memory of my grandmother, Allam Lendehee, and my wife's father, Houshang Dashti.

Acknowledgments

This is a great opportunity to express my respect for Dr. Kamy Sepehrnoori, my supervising professor, for his guidance, support, and providing me the opportunity to learn from him. He patiently guided me through the dissertation process, never accepting less than my best efforts. The knowledge and professional skills I learned from him will surely benefit me in my future career. I would also like to thank the other members of the supervising committe, Dr. Mojdeh Delshad, Dr. Gary Pope, Dr. Todd Arbogast, and Dr. Steven Bryant, for their time and comments.

I would like to express my sincere gratitude and appreciation to the members of GPAS team, past and present, for their contribution to the project. Special thanks go to Dr. Marcondes for helping me in development and implementation of the corner point method and full tensor formulation. I would like to express my gratitude to Dr. Chyoongyong Han for being a constant source of support and enthusiasm.

I am truly indebted to my professors, fellow students and administrative personnel in the Department of Petroleum and Geosystems Engineering. I would like to express my deep gratitude to my friends Kiomars Eskandari, Forouzan Zanjefili, and Javad Behseresht for their special help and support and to the members of my research group, Waleed Fazelipour, Abdoljalil Varavei, Peyman Pourafshary, Nariman Fathi, Mehdi Haghshenas, Mehdi Shirdel, Davood Ghrobani, and Reza Naimi-Tajdar It has been a pleasure to share unforgettable moments with them.

I am very grateful to Jana Cox for her wonderful help in editing my dissertation. Special thanks to Tim Guinn and Roger Terzian for management of the computer resources.

Special thanks go to Dr. Farhang Jalali and faculty and members of the Institute of Petroleum Engineering (IPE) of Tehran University for providing the financial support during my study.

I would like to express my most sincere gratitude and appreciation to my wife, Ghazal Dashti. This work could not have been accomplished without her patience, dedication and encouragement.

Farhad Tarahhom

The University of Texas at Austin
December 2008

Development of an Implicit Full-Tensor Dual Porosity Compositional Reservoir Simulator

Publication No. ____________________

Farhad Tarahhom, Ph.D.
The University of Texas at Austin, 2008

Supervisor: Kamy Sepehrnoori

A large percentage of oil and gas reservoirs in the most productive regions such as the Middle East, South America, and Southeast Asia are naturally fractured reservoirs (NFR). The major difference between conventional reservoirs and naturally fractured reservoirs is the discontinuity in media in fractured reservoir due to tectonic activities. These discontinuities cause remarkable difficulties in describing the petrophysical structures and the flow of fluids in the fractured reservoirs.

Predicting fluid flow behavior in naturally fractured reservoirs is a challenging area in petroleum engineering. Two classes of models used to describe flow and transport phenomena in fracture reservoirs are discrete and continuum (*i.e.* dual porosity) models. The discrete model is appealing from a modeling point of view,

but the huge computational demand and burden of porting the fractures into the computational grid are its shortcomings.

The affect of natural fractures on the permeability anisotropy can be determined by considering distribution and orientation of fractures. Representative fracture permeability, which is a crucial step in the reservoir simulation study, must be calculated based on fracture characteristics. The diagonal representation of permeability, which is customarily used in a dual porosity model, is valid only for the cases where fractures are parallel to one of the principal axes. This assumption cannot adequately describe flow characteristics where there is variation in fracture spacing, length, and orientation. To overcome this shortcoming, the principle of the full permeability tensor in the discrete fracture network can be incorporated into the dual porosity model. Hence, the dual porosity model can retain the real fracture system characteristics.

This study was designed to develop a novel approach to integrate dual porosity model and full permeability tensor representation in fractures. A fully implicit, parallel, compositional chemical dual porosity simulator for modeling naturally fractured reservoirs has been developed. The model is capable of simulating large-scale chemical flooding processes. Accurate representation of the fluid exchange between the matrix and fracture and precise representation of the fracture system as an equivalent porous media are the key parameters in utilizing of dual porosity models. The matrix blocks are discretized into both rectangular rings and vertical layers to offer a better resolution of transient flow. The developed model was successfully verified against a chemical flooding simulator called UTCHEM. Results show excellent agreements for a variety of flooding processes.

The developed dual porosity model has further been improved by implementing a full permeability tensor representation of fractures. The full permeability feature in the fracture system of a dual porosity model adequately captures the

system directionality and heterogeneity. At the same time, the powerful dual porosity concept is inherited. The implementation has been verified by studying water and chemical flooding in cylindrical and spherical reservoirs. It has also been verified against ECLIPSE and FracMan commercial simulators. This study leads to a conclusion that the full permeability tensor representation is essential to accurately simulate fluid flow in heterogeneous and anisotropic fracture systems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A large percentage of oil and gas reservoirs in the most productive regions such as the Middle East, South America, and Southeast Asia are naturally fractured reservoirs (NFR). The major difference between conventional reservoirs and naturally fractured reservoirs is the discontinuity in media in fractured reservoir due to tectonic activities. These discontinuities cause remarkable difficulties in describing the petrophysical structures and the flow of fluids in the fractured reservoirs.

Predicting the behavior of naturally fractured reservoirs is one of the most challenging areas in the petroleum engineering. Simulators must be developed to realize the effect of fractures on the overall reservoir behavior. The huge computational demand in naturally fractured reservoir simulations is a great obstacle. Numerous developments in computers and science empower us to use new methods to solve transport phenomena in porous media. The ability of solving large problems in reservoir engineering can be facilitated by using parallel processing and fast solvers. Development of efficient computational methods for solution of systems of linear and nonlinear equations in conjunction with use of parallel computers can enable us to simulate naturally fractured reservoirs much more efficiently.

To simulate and describe naturally fractured reservoirs, two classes of models have been developed; *continuum* and *discrete* models. Continuum models simplify a complex and irregular geometry system by characterizing several length scales. These methods can be used to describe phenomena in a macroscopic level. The second models, discrete models, have been used to describe phenomena in a microscopic level. The discrete models need large computational efforts, hence, they become less applicable. The dual porosity or dual permeability model, which is one of the continuum categories, is the most applicable model to simulate fractured reservoirs.

A new simulator, the General Purpose Adaptive Simulator (GPAS), is under development at the Center for Petroleum and Geosystems Engineering (CPGE) at The University of Texas at Austin to enhance modeling of oil recovery processes at large scales and high resolutions. GPAS is a fully-implicit, three-dimensional, multiphase and multicomponent compositional simulator and has the additional capability of parallel processing using Message-Passing Interface (MPI) libraries on distributed memory computers. The framework of GPAS performs the domain decomposition which namely divides the entire reservoir into several subdomains and distributes these subdomains among processors. Also, the framework automatically adds an additional layer to each subdomain. This layer represents the neighbored gridblocks which belong to other processors. The data in these communication layers will be updated by an update subroutine. Hence, each processor works only on one portion of the reservoir.

The current focus in GPAS is to improve the modeling of oil recovery processes by chemical methods. For conventional reservoirs in GPAS, chemical species properties are calculated explicitly and the effect of the phase behavior are coupled into the equation of state (EOS) model using a fully implicit implementation. A project has been underway at CPGE to develop a fully implicit chemical model for conventional reservoirs in GPAS.

The effect of natural fractures on the permeability anisotropy can be determined by considering distribution and orientation of fractures. Representative fracture permeability, which is a crucial step in the reservoir simulation study, must be calculated based on fracture characteristics. The diagonal representation of permeability, which is customarily used in dual porosity model, is valid only for the case that fractures are parallel to one of principal axes. This assumption cannot adequately describe flow characteristics where there is variation in fracture spacing, length, and orientation. To overcome these shortcomings in a dual porosity model, the principle of the full permeability tensor in the discrete fracture network can be incorporated into the dual porosity model. Therefore, the dual porosity model can closely retain the real fracture system characteristics.

A literature review of the characterization and properties of the dual porosity model and its features is presented in Chapter 2. A brief overview of GPAS and its routines is presented in Chapter 3. In Chapter 4 and 5, the governing equations, solution procedures, the mathematical and numerical formulations are presented. Chapter 6 covers the background, formulation, and implementation of a full permeability tensor representation of permeability in GPAS. Model verifications are presented in Chapter 7 and the parallel efficiency and speedup of the new simulator are presented in Chapter 8. Finally, the conclusions and recommendations are given in Chapter 9.

Chapter 2

Literature Review

Studies show the increase in the gap between oil production and oil global demand while a substantial amount of oil remains in the reservoirs and cannot be produced by conventional methods. Therefore, chemical enhanced oil recovery (EOR) plays an important role to fill the gap between production and the global demand. Great care must be taken in the design of an EOR process. A large percentage of oil and gas reservoirs are naturally fractured reservoirs with complex geometry. Understanding the physics of fluid flow in fractures is the key to better design EOR processes. Numerical simulators are substantial tools in designing a cost effective process and lowering the risk of failure.

The first part of this chapter explains geological features of naturally fractured reservoirs. Then, a literature review of numerical simulation of naturally fractured reservoirs along with a description of full permeability tensor is presented. The last part of this chapter is a literature survey of chemical phase behavior.

2.1 Basic Parameters

A fracture is defined as a surface where loss of cohesion occurs (Van Golf-Racht, 1982). Fracture properties are generally different than that of the host rock, and consequently lead to heterogeneity and anisotropy of flow. The study of naturally fractured reservoirs requires the examination of single-fracture parameters and multifracture parameters. Single-fracture parameters such as *aperture*, *size*, *nature*, and *orientation* are used to define the fracture by itself while multifracture parameters such as *arrangement*, *distribution*, *intensity*, and *density* are used to describe a group of fractures.

2.1.1 Single Fracture Parameters

Fracture aperture or fracture width refers to the perpendicular distance between fracture walls and directly affects fracture permeability. The fracture aperture may depend on depth, pore pressure and rock type. During the production where the overburden pressure remains constant while the pore pressure reduces due to reservoir depletion, the fracture aperture becomes smaller. Most of the time the difference between the fracture aperture at the reservoir condition and surface condition is due to the release of pore pressure of sample in surface condition. Fracture aperture is a key parameter in full permeability tensor calculation.

Fracture size is a representation of the fracture extension. Based on the size, fractures are divided into three categories; *Minor*, *Average*, and *Major* fractures. Minor fractures are fractures that have a length smaller than a single layer thickness. They have smaller aperture and often are filled. Average fractures cross more layers. Major fractures are very large fractures and often travel tens or more than hundreds of meters. They have large aperture and commonly are not filled or mineralized. A natural fracture often idealized as a penny-shape or a polygon-like shape.

The nature of fractures is a term describing the state of a fracture based on the opening, filling and wall characteristics. Fractures are opened or closed and they might be filled with a mineral or several minerals. Fracture walls can be frilly, smooth, polished or creeping.

Fracture orientation is a parameter which relates a single fracture to the environment. The fracture orientation is defined by two angels, dip azimuth δ and dip angle ω. In Cartesian Coordinates, the fracture orientation can be determined by three angles between the unit vector of the fracture and each axes. Fracture orientation contributes significantly to directional permeability and network connectivity.

2.1.2 Multifracture Parameters

Effective evaluation and production from naturally fractured reservoirs requires a deep understanding of the role of fracture networks on the fluid flow behavior. Albeit fractures are present at different scales in most reservoirs, their effect on fluid flow becomes important when they have enough spacing and connectivity.

A matrix block unit, or simply a matrix block, is formed from disjointing reservoir rock by fractures. Each matrix block is surrounded by fractures and may have any geometry. The block geometry is a primal factor in evaluating recovery from naturally fractured reservoirs . For practical purpose, simplified geometrical shapes, such as cubes or parallelepipeds, are used in naturally fractured reservoir models.

Fracture distribution is an indicator of fractures interconnectivity. This indicator is stronger if the fracture systems are highly connected, equivalent to each other, and can be seen as an indicator of fracture density and fracture intercommunication. The fracture distribution is a key factor in defining the shape, volume, and height of matrix blocks.

The term fracture density is used to define the degree of rock fracturing. The *volumetric fracture density* is used when the ratio is referred to as the bulk volume, or in other words, it is the ratio of cumulative fracture surface and matrix bulk volume. When the ratio relates to the area or to a length, it is called *areal* or *linear fracture density.* Unlike the linear and areal densities which are related to the direction of fluid flow, the volumetric fracture density is a stationary parameter (Van Golf-Racht, 1982).

Fracture intensity is defined as the ratio between fracture frequency and layer thickness frequency. It is an indication of fracture heterogeneity. The fracture intensity varies between 0.01 and 100, in which the bigger number shows stronger fractured zone.

2.2 Petrophysical Parameters

Petrophysical properties of fractures such as permeability and porosity differ from the host rock, and consequently introduce heterogeneity and anisotropy to flow. Shear fractures (i.e., faults) create barriers to flow by lowering the local permeability while extension fractures (*i.e.*, joins, microcracks) enhance the local permeability by producing conduits for flow. Characterization and the relation between fracture system and fluid flow can be addressed by considering the geometrical attributes of the fractures.

2.2.1 Porosity

Porosity determines the storage capacity for hydrocarbons. There are two kinds of porosity in fractured reservoirs: primary porosity and secondary porosity. The primary porosity is called the matrix porosity (ϕ_m), while the secondary porosity is called the fracture porosity (ϕ_f). The total porosity for a naturally fractured reservoir is simply the sum of the matrix and the fracture porosities. The fracture porosity is remarkably lower than the matrix porosity. From the storage point of view, the accuracy of the determination of fracture porosity seems unimportant, but it has significant effects on the transient flow.

2.2.2 Permeability

Permeability determines the fluid flow capacity of the rock. Analogous to porosity, there are two types of permeability in fractured reservoirs: fracture permeability and matrix permeability. The fracture permeability may be defined as single fracture permeability, fracture network permeability, or fracture permeability of fracture-bulk volume. For a single fracture which forms an angle α with the flow direction, the fracture permeability is defined by (Van Golf-Racht, 1982)

$$K_f = \frac{b^2 \cos^2 \alpha}{12} \tag{2.1}$$

where K_f is the fracture permeability and b is the fracture opening. For a fracture system with n fractures having the same orientation, the fracture permeability is expressed by

$$K_f = \frac{\cos^2 \alpha}{12} \sum_{i=1}^{n} b_i^2 \tag{2.2}$$

For a set of fracture systems (a fracture network) with different orientation for each system, the fracture permeability is expressed by

$$K_f = \frac{1}{12} \left[\cos^2 \alpha \sum_{1}^{n_\alpha} b_{\alpha i}^2 + \cos^2 \beta \sum_{1}^{n_\beta} b_{\beta i}^2 + \cdots \right] \tag{2.3}$$

where α and β are fracture systems with different orientations. The above permeability is called *intrinsic fracture permeability* because the cross-sectional area in the calculation is the fracture cross-sectional area. In other words, the intrinsic fracture permeability ignores the rock bulk volume. Based on Darcy's law, the cross-sectional area of the rock bulk volume should be used in calculations. Consequently, the fracture permeability must be tuned based on the rock bulk volume cross-sectional area. The total permeability can be expressed by

$$K_t = K_m + K_f \frac{\sum_{i=1}^{n} b_i}{h} \tag{2.4}$$

where K_t is the total permeability and h is the thickness of the rock bulk volume. Predicting the fluid flow behavior of naturally fractured reservoirs is very difficult due to the complexity of the fracture systems. Fractured reservoirs are reservoirs with extreme contrasts between high permeability zones and low permeability zones. Simple geometric models have been developed to deal with this complexity (Reiss, 1982). These models, which are classified as *dual porosity* models, consist of matrix blocks separated with highly interconnected sets of parallel fractures in two or three dimensions (Fig. 2.1).

2.3 Numerical Simulation of Fractured Reservoirs

Analytical solutions of fluid flow in fractured systems are only applicable for very simple cases, such as one-dimensional or radial flow of single or two phases. Numerical simulators are useful tools for maximizing oil or gas recovery by studying different mechanisms in complicated fractured reservoirs. These simulators need to account for the complex geometry of the fractures and to consider all the relevant mechanisms, especially the transfer of fluids between fracture and matrix.

To simulate and describe naturally fractured reservoirs, two classes of models have been developed; *continuum* and *discrete* models. Continuum models (i.e. dual porosity) simplify a complex and irregular geometry system by characterizing several length scales. These methods can be used to describe phenomena on a macroscopic level. The second models, discrete models, have been used to describe phenomena on a microscopic level.

2.3.1 Dual Porosity Models

To obtain the effective properties of porous media, an average should be taken over a volume which is small in comparison with volume of the system, but large enough to apply the equation of change to that volume. Because the structure of porous media is so complicated, speculation of effective properties is somewhat difficult even though the averaging method makes sense. To obtain the effective transport coefficient, many efforts have been made to simplify representation of porous media. These methods are refered to continuum models.

The dual porosity model was developed by Barenblatt *et al.* (1960) and was enhanced and introduced to the petroleum reservoir by Warren and Root (1963) and Odeh (1965). The dual porosity model envisions the porous system as two distinct continua; the fracture with low storage capacity but high flowing capacity, and the matrix with high storage capacity and low flowing capacity. An idealized model of highly interconnected fractures which are fed by numerous matrix blocks is shown in Fig. 2.1. By using the equation of motion and conservation of mass for each medium separately, and a source or sink term for the transfer of fluid between the two medium, Barenblatt *et al.* (1960) derived the following equations for a dual porosity system:

$$\begin{cases} \frac{K_1}{\mu}\nabla P_1 = \Phi_1 C_1 \frac{\partial P_1}{\partial t} - \varphi_1 \frac{\partial P_2}{\partial t} + \frac{SK_1}{\mu}(P_1 - P_2) \\ \frac{K_2}{\mu}\nabla P_2 = \Phi_2 C_2 \frac{\partial P_2}{\partial t} - \varphi_2 \frac{\partial P_1}{\partial t} + \frac{SK_1}{\mu}(P_1 - P_2) \end{cases} \tag{2.5}$$

where the indices 1 and 2 refer to matrix and fracture systems respectively and

K = Permeability, L^2
P = Pressure, $\frac{M}{Lt^2}$
C = Compressibility, $\frac{Lt^2}{M}$
μ = Viscosity, $\frac{M}{Lt}$
S = Fracture and matrix contact surface, L^2

The parameters φ_1 and φ_2 are relating the change in porosity of matrix and fracture systems with the change in fluid pressures P_1 and P_2.

The classical continuum equation of transport is used to describe the fluid flow behavior in each system. The two systems can exchange fluids at their interfaces. In the Warren and Root's model, the matrix blocks are homogeneous and isotropic. The fractures are uniform and parallel to one of the principal axes. Flow cannot occur between matrix blocks, and the quasi-steady state exists in the matrix block.

The difference between Barenblatt and Warren and Root solutions is that Barenbalett neglected fracture and rock compressibilities, while Warren and Root considered the fracture compressibility but ignored the flow through the blocks. The corresponding equation for Warren and Root approach are as follow
Fracture system:

$$\frac{K_{fx}}{\mu}\frac{\partial^2 P_f}{\partial x^2} + \frac{K_{fy}}{\mu}\frac{\partial^2 P_f}{\partial y^2} - \phi_m C_m \frac{\partial P_m}{\partial t} = \phi_f C_f \frac{\partial P_f}{\partial t} \tag{2.6}$$

Matrix system:

$$\phi_m C_m \frac{\partial P_m}{\partial t} = \frac{\sigma K_m}{\mu}(P_f - P_m) \tag{2.7}$$

where

K_{fx} = Fracture permeability in x direction, L^2

P_f = Fracture pressure, $\frac{M}{Lt^2}$

μ = Viscosity, $\frac{M}{Lt}$

K_{fy} = Fracture permeability in y direction, L^2

P_m = Matrix pressure, $\frac{M}{Lt^2}$

C_f = Fracture compressibility, $\frac{Lt^2}{M}$

C_m = Matrix compressibility, $\frac{Lt^2}{M}$

ϕ_f = Fracture porosity, fraction

ϕ_m = Matrix porosity, fraction

The parameter σ in Eq. 2.7 depends on the geometry of the matrix blocks and is expressed by

$$\sigma = \frac{4n(n+2)}{L^2} \tag{2.8}$$

where n is the number of fractures ($n = 1, 2, 3$) and L is the characteristic length of the block given by

$$\begin{aligned} L &= a && \text{for } n = 1 \\ L &= \frac{2ab}{a+b} && \text{for } n = 2 \\ L &= \frac{3abc}{ab+bc+ca} && \text{for } n = 2 \end{aligned} \tag{2.9}$$

where a, b, and c are the lengths of the block faces. Warren and Root derived an analytical solution for Eq. 2.6 and Eq. 2.7 in the following form:

$$P_D(t_D) = \frac{1}{2}\left[0.80908 + \ln t_D + Ei\left(-\frac{\lambda t_D}{\omega(1-\omega)}\right) - Ei\left(-\frac{\lambda t_D}{1-\omega}\right)\right] \tag{2.10}$$

where Ei is the exponential integral function defined by

$$Ei(-x) = -\int_x \frac{e^{-u}}{u} \mathrm{d}u \tag{2.11}$$

The parameter λ depends on the shape factor and the ratio of fracture permeability to the matrix permeability and ω is the ratio of fracture storage capacity (ϕC) to the total storage capacity (sum of the fracture and matrix storage capacities).

$$\begin{aligned} \lambda &= \frac{\alpha K_1 r_w^2}{K_2} \\ \omega &= \frac{\Phi_2 C_2}{\Phi_1 C_1 + \Phi_2 C_2} \end{aligned} \tag{2.12}$$

Equation 2.10 is an special solution of Eqs. 2.6 and 2.7 at the well location, and is applicable in well test analysis or near wellbore problems.

Kazemi (1969) developed a single phase dual porosity model for a two dimensional radial system. To simplify the problem, the following assumptions were made:

- Matrix blocks have a high storage capacity and a low flowing capacity while the fractures have a low storage capacity and a high flowing capacity.
- The flow takes place only through fractures.
- The matrix and fractures are homogeneous and isotropic and the reservoir is horizontal.
- Un-steady state flow conditions.

Also, for simplicity only, a single fracture which was a representative of all horizontal fractures was used. The gravity term was ignored when the equations were solved. The model was compared with the Warran and Root solutions for pressure drawdown and pressure build-up tests. The model was able to reproduce the same results as Warren and Root except for the transition zone. The use of a transient transfer function deviated the results from the Warren and Root model, which considered a quasi-steady state transfer function.

Kazemi *et al.* (1976) developed a more advanced model for multiphase systems. A quasi-steady source or sink function corresponding to the potential difference between the fracture and the matrix were used in this model. They derived two distinct sets of equations, one for the matrix system and another for the fracture system, which is connected by a so-called transfer function as follow

Fracture system:

$$\nabla \lambda_{\alpha f} \left(\nabla P_{\alpha f} - \rho_\alpha \frac{g}{g_c} \nabla D \right) = \frac{\partial}{\partial t} \left(\frac{\phi S_\alpha}{B_\alpha} \right) + q_{\alpha f} + \tau_{\alpha m-f} \tag{2.13}$$

Matrix system:

$$\tau_{\alpha m-f} = \frac{\partial}{\partial t} \left(\frac{\phi S_\alpha}{B_\alpha} \right)_m \tag{2.14}$$

transmissibility, $\lambda_{\alpha f}$, is defined as

$$\lambda_{\alpha f} = \frac{Kr_{\alpha f}}{\mu_\alpha B_\alpha} K_f \tag{2.15}$$

with

B = Formation volume factor, $\frac{L^3}{\mathrm{std}L^3}$
g = Gravitational acceleration, $\frac{L}{t^2}$
g_c = Gravitational conversion factor, $\frac{ML}{t^2F}$
K = Permeability, L^2
Kr = Relative permeability, dimensionless
P = Pressure, $\frac{M}{Lt^2}$
q = Production or injection rate, $\frac{L^3}{t}$
τ_{m-f} = Matrix-fracture transfer function flow, $\frac{L^3}{t}$
S = Saturation, fraction
t = time, t
D = Depth measured positive downward, L
μ = Viscosity, $\frac{M}{Lt}$
ρ = Density, $\frac{M}{L^3}$
ϕ = Porosity, fraction

subscripts
f = Fracture
m = Matrix
α = Phase

Kazemi *et al.* (1976)'s transfer function, $\tau_{\alpha m-f}$, can be written in a general form (Chen, 1993):

$$\tau_{\alpha m-f} = -T_{\alpha m-f}\left[\left(P_{\alpha f} - \rho_{\alpha f}\frac{g}{g_c}D_f\right) - \left(P_{\alpha m} - \rho_{\alpha m}\frac{g}{g_c}D_m\right)\right] \tag{2.16}$$

$T_{\alpha m-f}$ is the matrix-fracture transmissibility and is defined as

$$T_{\alpha m-f} = \sigma V_b K_m \left(\frac{Kr}{\mu B}\right)_{\alpha m} \tag{2.17}$$

The shape factor is defined as

$$\sigma = 4\left(\frac{1}{L_x^2} + \frac{1}{L_y^2} + \frac{1}{L_z^2}\right) \tag{2.18}$$

Gilman and Kazemi (1983) extended Kazemi *et al.* (1976)'s model to simulate polymer flooding and tracer transport. They added a weight fraction of chemicals in the fracture systems and matrix blocks.

Rossen (1977) extended a conventional single porosity model based on the Kazemi *et al.* (1976) idealized model. The transfer of fluid to and from the matrix block were treated as a production or injection. Transfer of fluid from matrix to fracture was representated as a *source* term and the transfer of fluid from fracture to matrix was representated as a *sink* term. The transfer functions were handled semi-implicitly. The advantage of treating transfer functions semi-implicitly was to improve the stability of the system.

The distinction between most of the dual porosity models is in the definition of the transfer function. Many attempts have been made to incorporate gravity and capillary effects into the transfer function. The effect of gravity on the transfer of fluid between matrix and fracture were initially discussed by Reiss (1980). Litvak (1985) suggested a transfer function to consider the capillary forces and gravity effects as well as the pressure gradient between matrix blocks and surrounding fractures:

$$\tau_{\alpha m-f} = -T_{\alpha m-f}\left(P_m - P_f\right) + CG_{\alpha m-f} \tag{2.19}$$

where

$$CG_{\alpha m-f} = T_{\alpha m-f} S_{\alpha f}\left[(\rho_w - \rho_o)\frac{g}{g_c}(L_z - Z_{\alpha m}) + Pc_{owm}\right] \tag{2.20}$$

and L_z is the matrix block height, $Z_{\alpha m}$ is the height of the fluid α in the matrix block, $S_{\alpha f}$ is the fracture saturation of phase α, and Pc_{owm} is the oil-water capillary pressure in the matrix block. The fracture capillary pressure assumed to be zero. $S_{\alpha f}$ is used to account for partially immersed matrix blocks.

Thomas *et al.* (1983) developed a 3D, three-phase, finite difference dual porosity model for simulating naturally fractured reservoirs. They used the pseudo-capillary pressure and relative permeability approach to account for the gravity drainage. The integration of pseudo-curves reduces the three-dimensional problem to a two-dimensional problem. They simulated hysteresis effects for both capillary pressure and relative permeability. No matrix subgridding was done in their model (See Section 2.3.2 on page 18 for more information). The pseudo-techniques were also used by Dean and Lo (1986), and Rossen and Shen (1987) to simulate naturally fractured reservoirs.

Beckner *et al.* (1987) also used the pseudo-capillary pressure to simulate a single matrix block in a fracture network. They used the experimental data of Kleppe and Morse (1974) to verify their model. They claimed that using a transfer function based on the matrix flow properties is not enough to predict the multidimensional fluid exchange between matrix and fracture. They also mentioned that the shape factor is a time dependent parameter, and multiple shape factors are required to model a waterflooding process with different injection rates. They introduced a dynamic imbibition transfer function to overcome this problem.

In a dual porosity model, it is assumed that all matrix blocks within a fracture have the same average pressure and saturation. Using the averaged matrix water saturation to calculate the capillary gradient between matrix block and fracture will result in an inaccurate pressure gradient between matrix and fracture. Also, the gravity effect cannot be accounted for with a single matrix node. Subgridding the matrix block is a good approach to accutately model the transient flow in matrix blocks.

2.3.2 Subgridding Techniques

Saidi (1983) improved the dual porosity model by introducing a new concept called subgridding. Saidi developed a three-phase, three-dimensional dual porosity simulator. Saidi discretized the matrix blocks in the vertical and radial directions, see Fig. 2.2. The model was a cylindrical model and the matrix blocks were discretized horizontally and vertically. This was a big improvement to account for the gravity drainage when the matrix permeability is very low. Chen and Fitzmorris (1987) used the same approach in their thermal simulator for fractured reservoirs.

Pruess and Narasimhan (1985) introduced a “multiple interaction media” (MINC) method. Since the changes in thermodynamic conditions are very small in the direction of fracture than perpendicular to it, they assumed that the surfaces with the equal distance from the fracture have the same potential. Therefore, they discretized the matrix such that all interfaces between volume elements are parallel to the nearest fracture. The authors stated that the MINC method overestimates thermodynamic parameters at some part of the surface and underestimates them at the other parts. However, the total flow rate across interfaces at a constant distance from the fracture are accurate within a fraction of percent. A favorable situation is when there are uniform initial conditions in the matrix blocks and when the transient changes in fractures occur such that the matrix blocks are subject to approximately uniform boundary condition at all times.

Wu and Pruess (1986) used the MINC method to study the oil-water imbibition recovery. They compared the MINC method results against the fine-grid simulation results and concluded that the MINC method accurately predicts the water imbibition. They also mentioned that the use of basic transfer function yield in large errors for simulation of oil recovery from a single matrix block. The error increases rapidly with increasing the matrix block size or fluid viscosity, and with the decrease in the matrix permeability.

Gilman (1986) developed a dual porosity simulator based on the MINC method. Gilman divided the matrix block into a rectangular and vertical subdomains (Fig. 2.3). A shape factor was used to calculate flow between the matrix and fracture. This method has the same advantages as Saidi's model in terms of gravity drainage. The number of variables increases from four per each gridblock to $2(N_m + 1)$, where N_m is the number of matrix subgrids. The solution of the system becomes the major problem in dual porosity model with subgriddings. Gilman used a rigorous technique to separate the fracture equation from the matrix equation to reduce the solution time. In the solution of the system, he decoupled matrix equation from fracture equation, while solving the whole system implicitly.

Beckner *et al.* (1991) used a subdomain method to develop a 3D, three-phase, dual porosity simulator. Similar to Gilman (1986), they discretized the matrix blocks into rectangular rings and layers. The big improvement was the use of the geometrical aspects of the subdomain grid in calculation of subdomain face transmissibility. Instead of using a shape factor, they calculated a transmissibility at each face using matrix geometry. The geometrical aspects include cross sectional area, directional absolute permeability and distance between subdomain centers.

Chen (1993) developed a new dual porosity simulator, UTDUAL, for modeling waterflooding in fractured reservoirs. To minimize coding, Chen decoupled the matrix-block pressure equation from the fracture pressure equation. He studied countercurrent imbibition processes in naturally fractured reservoirs.

Aldejain (1999) implemented a dual porosity model into a chemical flooding simulator, UTCHEM. He used an implicit pressure explicit concentration method (IMPEC) to solve the matrix and fracture systems.

Naimi-Tajdar *et al.* (2007) implemented a generalized dual porosity model, the multiple-interacting-media (MINC), into a General Purpose Adaptive Simulator (GPAS). To better describe transient flow in matrix blocks, he discretized matrix blocks into both horizontal and vertical directions.

2.4 Discrete Fracture Models

Unlike the continuum model, which uses the averaging methods over the volume of the system; the discrete fracture network (DFN) model gives a realistic picture of the reservoir. This method depicts the fracture connectivity completely differently than other methods.

In continuum methods, such as dual porosity or dual permeability, the matrix and fracture are represented as two different systems. In general, in continuum models permeabilities of the gridblocks are represented as diagonal tensors. Hence, these methods do not depict the fluid flow pathway especially when the fracture discontinuity has occurred. Another shortcoming is the complexity in calculation of transfer functions between fracture and matrix blocks.

The discrete fracture model is an alternative to enhance the fractured reservoir simulators. In this model, each fracture is modeled as one or more 1D, 2D or 3D elements. All of the properties such as transmissibility, storage, size, and orientation are assigned to each fracture. The fractures are generated deterministically, which are detected through seismic data, or stochastically, which are generated by a stochastic method.

Discrete fracture models usually require an unstructured discretization scheme. There are two main unstructured discretization approaches: finite-element and finite volume (or control volume finite-difference) methods. Noorishad and Mehran (1982) and Baca *et al.* (1984) used a finite-element method to simulate a single-phase flow in fractured reservoirs. The dimension of the fracture is reduced from n to $(n-1)$ in discrete fracture models which in turn reduces the computational works. Kim and Deo (2000) and Karimi-Fard *et al.* (2003) extended Baca *et al.* (1984)'s model for two-phase flow. They used a superposition method to couple matrix and fracture systems. In their methods, they discretized the matrix and fracture separately and used a superposition method to add the contribution of each system on the overall flow. Hoteit and Firoozabadi (2006a,b) used a mixed finite-element (MFE) method and the discontinuous Galerkin (DG) method to solve the highly nonlinear flow equations. They used the MFE to solve the pressure equation implicitly and the DG to solve the mass balance equations explicitly.

The finite-element methods are successful in single-phase flow simulation, but they do not satisfy the local mass conservation in the case of multi-phase flow in highly heterogeneous fracture systems. Using a mixed or discontinuous Galerkin method resolves the problem, but these methods are more expensive than finite-volume methods (Karimi-Fard *et al.*, 2004).

The control volume finite element and finite element methods are very similar in a sense that the same type of interpolation functions can be used for dependent variables in both methods. The difference between the two is the calculation of flux between control volumes. In the finite element method the fluid potentials are calculated independently of the fluxes between nodes, while in the control volume finite element method the fluid flux is calculated explicitly and mass balance is calculated from the fluxes (Fu *et al.*, 2005).

Baliga and Patankar (1980) were among the first authors to propose a control-volume (CV) method in fluid dynamics. The advantages of CV over MFE are: 1) local mass conservation is satisfied; 2) choosing the upwind point is based on the flow potential at the boundaries which has a clear physical meaning; and 3) it can be used for hyperbolic and convection-diffusion partial differential equations. Due to these capabilities, the CV is a method of choice for solving the saturation equation of two immiscible fluids.

Bastian *et al.* (2000) used the CV method for simulating two-phase flow in fracture systems. In their work, the fracture and matrix capillary pressure were equal. Geigar *et al.* (2003) also developed a numerical simulator for fractured systems based on the CV method. They used an FEM to solve the pressure equation and CV for solving the saturation equation. Despite the importance of capillary pressure, they did not include it in their model.

Monteagudo and Firoozabadi (2004) employed the CV method to develop a numerical simulator for a two-phase immiscible and incompressible flow in discrete-fractured media. They introduced a new 3D formulation for fractured media. Figure 2.4 shows a 2D Delauny triangulation of a 2D matrix and 1D fracture used in their model. The thick line in the figure is a 1D fracture which is divided into several segments. These segments are the edges of the Delauny triangles enclosing the 1D fracture. They assumed that the flow potentials are the same at the matrix and fracture interface.

The discrete model is appealing from a modeling point of view, however, the huge computational demand and burden in porting the fractures into the computational grid are its shortcomings.

2.5 Parallel Reservoir Simulation

Parallel processing is a technique to speedup the execution of a program by dividing the program into multiple fragments that can execute simultaneously, each on its own processor. A program being executed across n processors might execute n times faster than it would using a single processor. In practice, it is often difficult to separate a program in several fragments such that CPUs can execute different fragments without interfering with each other. The earliest computers could only run one job at a time. Even input/output (I/O) was being done in serial. An early form of parallel processing allowed the computer to start I/O while executing the process-intensive program.

The next achievement was multiprogramming. In multiprogramming, multiple programs were allowed to be submitted by the user. Each program used the processor for a short time and it seemed that all program were running at the same time. The problem with multiprogramming was the competition for the same resource at the same time by two or more programs which leads to the critical section runtime.

Vector processing was another step used to improve the performance by performing more than one operation at a time. In this case, the processors were designed such that they allowed a program to run mathematical operations on multiple data elements simultaneously. This method worked well in engineering applications where data naturally occurred in the form of vectors or matrices.

Multiprocessing was the next step in parallel processing. With single-CPU computers, it is possible to create a network of computers called a cluster to perform parallel processing. In this system, one or more processors share resources to do the job. The first generation had the master/slave configuration where the master was responsible for initiating all commands and controlling slaves. A symmetric multiprocessing system (SMP) was developed to allow each processor equally capable and responsible for managing the flow of work through the system. Distributed processing software is required to perform this type of parallel processing. Three major parallel processing techniques are: shared memory architecture (SMA), single instruction multiple data (SIMD), and multiple instruction multiple data (MIMD).

In shared memory architectures each processor has access to the content of a global memory through an interconnection network. The interconnection network provides a route for each processor to access the global memory. The global memory must be large and fast enough to support fast processors. The advantage of shared memory architectures is that processes executing in parallel can share data for a variety of purposes.

The SIMD computers were among the first system to provide computational power above the GFLOPS range. Any machine with the ability of processing many data items simultaneously by the same data manipulation is classified as SIMD machine. The disadvantage of SIMD is that all processors execute exaclty the same instruction at any instant.

MIMD architecture has several processors such that each processor executes independent sequence of machine instructions. In this technique, any processor executes the instruction by any accessible data rather than working only on a single data. If the system does not have shared memory (*i.e.*, distributed memory clusters), any data needed by more than one processor must be sent from processor to processor explicitly through the interconnection network.

Coding is the biggest challenge to the distributed MIMD. A massive amount of work is required to change the existing serial codes to work on MIMD machines. Programmers have to create the parallel instructions explicitly. Big problems with parallelism are: correctness of the results, reproduciblilty of the results, performance predictability, load balancing, synchronization and communication time (Kohar and Killough, 1995).

Most reservoir simulations are based on the finite difference method. Consequently there is communication between neighboring gridblocks for the flux or transport term. The PVT and rock properties calculations depend only on the gridblock data and do not need any communication. Load balancing and the message passing time are key parameters affecting the performance of a parallel computing. They lower the speedup when the number of processors increases.

Nolen and Patricia (1981) employed the vectorization technique in a fully implicit black-oil simulator. They reported a reduction in the scalar run time by a factor of 3.7 for a problem with a size of $25 \times 20 \times 4$ gridblocks. Except for the well performance and output routines, all other parts of code were implemented in a vectorized form. They could improve the performance of a fully implicit method to be only 50-80% slower than IMPES method per time-step. Since the fully implicit methods are capable of taking larger time-step, the overal run time could be smaller than IMPES methods.

Chien *et al.* (1987) investigated the feasibility of both vectorization and parallel processing on the speedup of a general purpose reservoir simulator. All calculation except the linear solver equation were implemented to facilitate vectorization and parallel processing. They reported the speedup ranges from 5.3 to 10, due to vectorization, and 3.3 on top of that due to parallel processing. Scott *et al.* (1987) showed a method for applying MIMD computers to reservoir simulation. They investigated the parallelization of the matrix coefficient and linear solver for a black oil model. Chien and Northrup (1993) presented the parallel processing on shared memory computers. Application of parallel processing on field cases have been addressed in many publications (Wallis *et al.*, 1991; Killough and Bhogeswara, 1991; Killough *et al.*, 1997; Dogru *et al.*, 1999; Liu *et al.*, 2000; Fjerstad *et al.*, 2005).

Al-Shaalan *et al.* (2003) developed a parallel dual porosity simulator for a multi-million gridblock fractured reservoir problem. They used a mixed approach with message passing interface (MPI) and open message passing (OpenMP). The OpenMP pattern was used for shared memory parallelism while the MPI pattern was used for the distributed memory parallelism. They could run a dual porosity dual permeability case study with 2.57 million gridblocks and sixty years of history in one-day turn around on the PC Xeon Cluster using 54 CPUs.

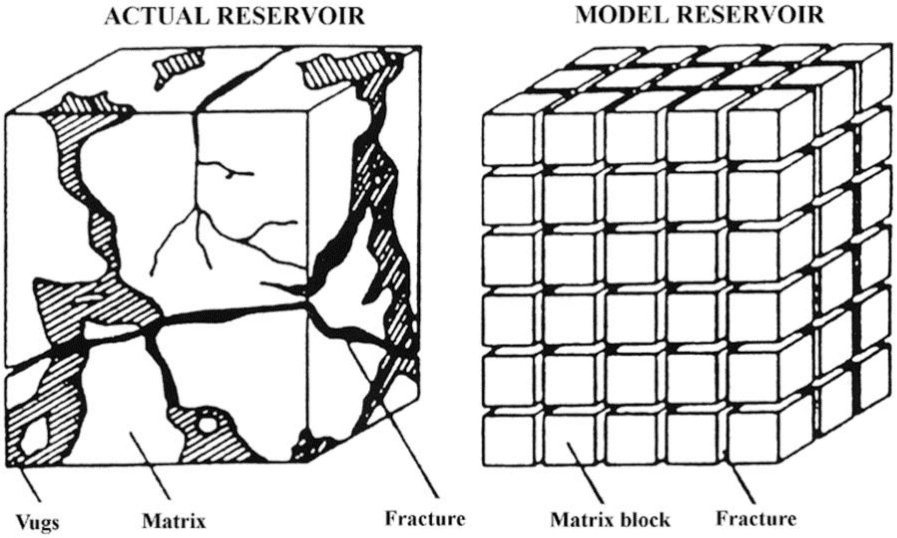


Figure 2.1: Idealization of a fractured system (Warren and Root, 1963)

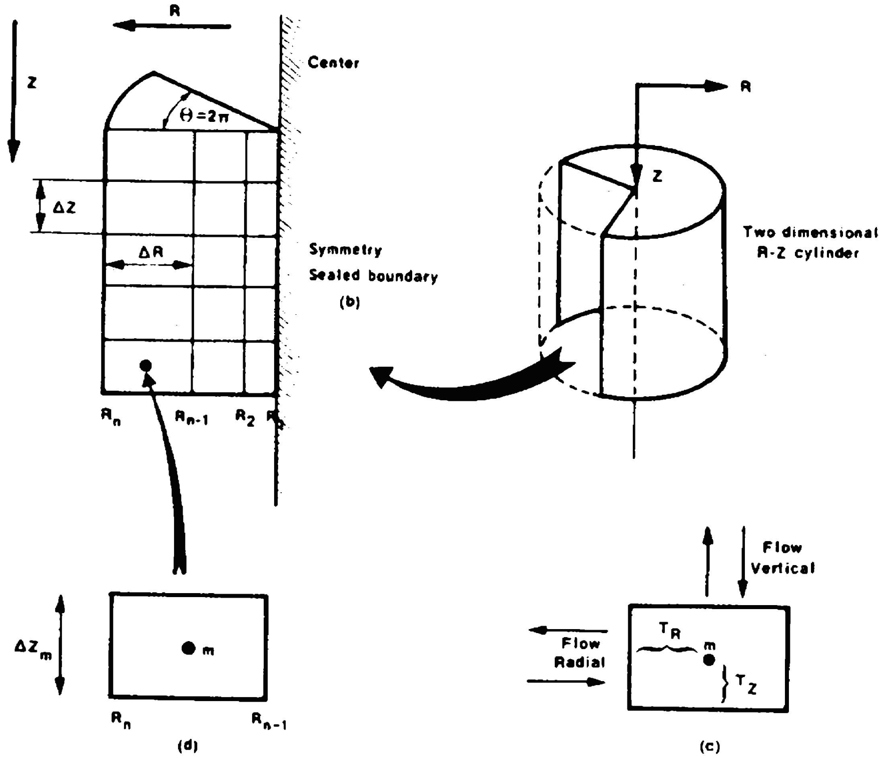


Figure 2.2: Matrix elements with gridblocks definition (Saidi, 1983)

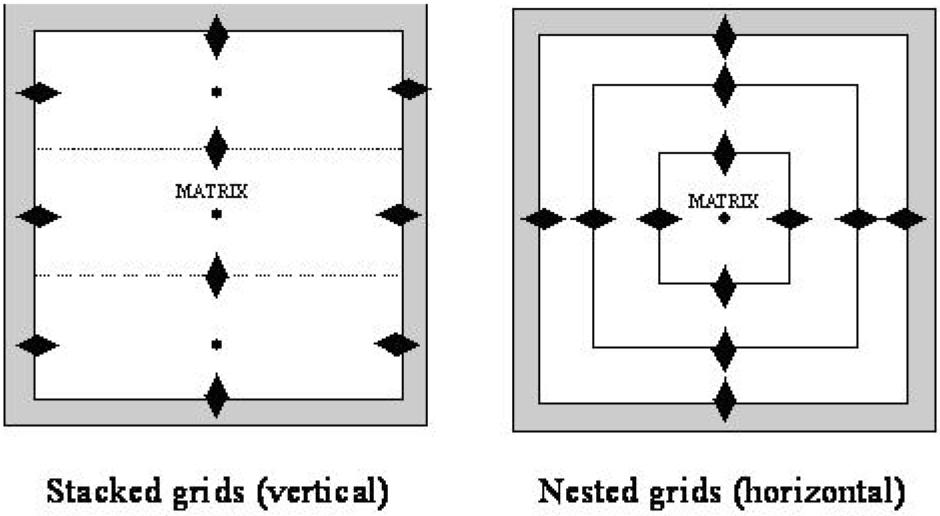


Figure 2.3: Subgrids of a single matrix block (Gilman, 1986)

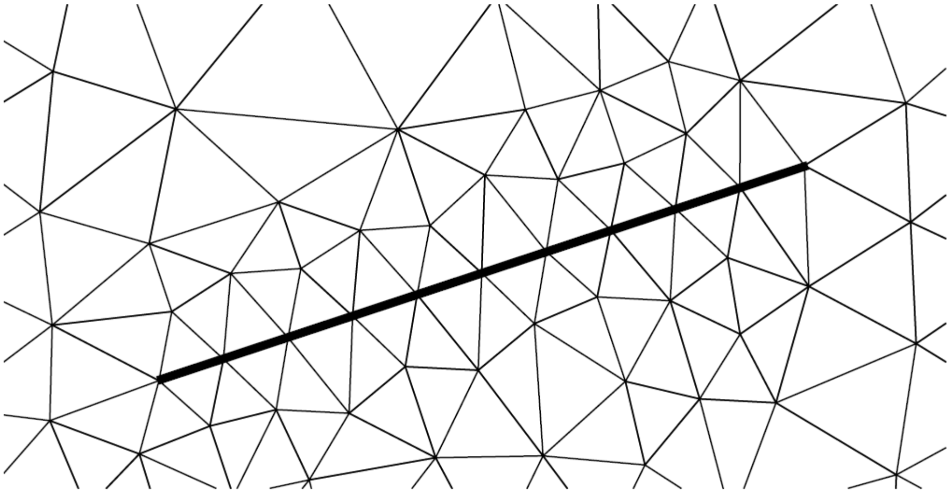


Figure 2.4: 2D Delauny triangulatioin conforming to a 1D fracture (Monteagudo and Firoozabadi, 2004)

Chapter 3

General Purpose Adaptive Simulator (GPAS)

Equation of state compositional reservoir simulators do not handle the modeling of aqueous phase behavior and those which are designed for chemical flood modeling typically assume simplified hydrocarbon phase behavior. There is a need to have a single reservoir simulator capable of combining both approaches to benefit from the advantages of both models. The overall objective of GPAS is to develop such technology using a computational framework that also allows parallel processing.

GPAS is a parallel, 3D, fully implicit, equation of state and chemical compositional simulator. The simulator is capable of performing accurate, efficient high-resolution simulation of fluid flow in porous media for large complex problems. Initially the code was developed for a compositional miscible gas flooding process but, was further developed for chemical processes in both conventional and naturally fractured reservoirs. An important aspect of GPAS is to develop a simulator to be able to economically solve multi-million gridblock problems and improve grid resolution.

To better cope with the complicated tasks associated with the parallel processing, the physical models are separated from the parallel processing. The reason for separating parallel processing from the physical model is to simplify the development of each model at the same time by different team members. The Integrated Parallel Accurate Reservoir Simulation (IPARS) framework (Parashar *et al.*, 1997; Wang *et al.*, 1997) is used to separate the parallel processing from the physical models. Several physical models have been developed under the IPARS framework (Wheeler *et al.*, 1999; Wang *et al.*, 1999; Nalla, 2002; John, 2003; Naimi-Tajdar *et al.*, 2007). The equation of state and chemical compositional dual porosity models also run under the IPARS framework.

3.1 Framework Description

IPARS was developed as a framework for single and parallel reservoir simulation research. The code for this framework is written for distributed memory, message passing computers. EOSCOMP (Wang *et al.*, 1997, 1999) is one of several models developed under the IPARS framework. Abate *et al.* (1999) tested the EOSCOMP and several other models on clusters of PCs and reported an excellent scaling on large numbers of processors running in parallel.

3.2 Framework Features

The IPARS framework supports three-dimensional multiphase, multicomponent flow. The framework includes a number of advanced features such as input processing, memory allocation and management, domain decomposition, well management, output generation and other functions like table lookup and interpolation. The framework handles heterogeneous reservoirs with variable porosity and permeability. Permeability was a diagonal tensor, but in this work we have added the ability to handle a full tensor permeability.

On multiprocessor machines, the grid system is distributed among the processors such that each processor is assigned a subset of the total grid system. The subgrid assigned to a processor is surrounded by a communication layer of grid elements that have been assigned to other processors. Several features of the framework in conjunction with the compositional models are:

- **Input/Output**. The framework allows each processor to read the entire input file and process the portion that is assigned to it. The outputs are collected by the master processor and written to the output files.

- **Domain Decomposition**. The reservoir domain is divided vertically (over the y-direction) into several subdomains equal to the number of processors allocated for the run. The information of each subdomain is distributed to one processor. Each processor reads the input file and collects the information about the subdomain belongs to it. An additional layer of gridblocks (ghost layer or communication layer) which physically surrounds the subdomain and belongs to other processor is added to each subdomain assigned to the processor. The framework provides a routine to update the data in the communication layers during the run time via a message passing interface (MPI). The framework supports an arbitrary number of wells with one or more completion intervals. On parallel runs, the well blocks may be assigned to more than one processor. For each processor, the framework computes the permeability, the geometrical constant in the productivity index, and the length of the open interval for each well.

- **Memory Management**. The framework is designed to be portable. FORTRAN77 and classical C code are used. The C code is used mainly for the memory management and user interaction. The framework allocates memory for the grid-element arrays. The first three dimensions of the array are used for the grid related properties. Extra indices are used for the phase and component information. The memory is allocated by a C subroutine which returns a pointer to the location of the variable in the memory. To allocate the memory, developers need to create an ID for the variable and define the size of the extra dimensions. The variables are stored in a common block for use in different routines. The framework supports two types of arrays; the arrays that are independent of the gridblock locations such as binary interaction coefficients and critical properties, and the grid-element arrays such as porosity and permeability.

- **Message Passing between Processors**. The message passing interface (MPI) is used in the framework for the communication between processors. The framework provides several routines to collect and distribute the physical properties such as pressure, saturation and residuals. To update a variable in the communication layers, the developer needs to provide the pointer to the location of the variable in the memory. In the solution of the linear equations, PETSc (refer to Section 3.5 on page 39 for more details) handles the communication between processors.

Several other calculations needed by physical models are also handled by the framework. These calculations include

- The calculation of the constant part of the transmissibilities
- The identification of the ghost layers
- Extracting the relative permeability and capillary pressure and computing their derivatives using a table lookup routine
- Identifying the well locations
- Calculating the productivity index for each gridblock of a well

The framework has an internal set of units for each physical quantity. The user can change the default unit by providing the correct physical unit in a pair of brackets. The framework converts the unit to its internal unit using a correct unit conversion factor.

The framework also provides several two-dimensional and N-dimensional utilities for interpolation and extrapolation of data. The two-dimensional utilities include:

1. Piecewise constant
2. Piecewise linear
3. Quadratic spline with optional pole
4. Cubic spline with optional pole
5. User-defined function

N-dimensional utilities include:

1. Bilinear interpolation (rectangular subdomain)
2. Quadratic interpolation (rectangular subdomain)
3. User-defined function

At the run time, the IPARS framework sets the number of processors requested by user by calling the `SETPRC` subroutine. The `SETPRC` subroutine sets multiprocessor parameters including number of processors, processor number, and process ID. For a single processor run, this routine sets the parameter of the master processor only. The IPARS framework then reads the entire input file and creates a long vector of all input keywords for later analysis. This vector is accessible to several routines to extract keywords and their allocated parameters. The scalar

keywords are extracted from the keyword super array by the subroutine `GETIDAT`. The number of gridblocks, number of phases, and the number of components are read to perform the memory allocation for the grid-element arrays. Then, based on the finite different method (7 point-stencil, 19 point-stencil, or 27 point-stencil), the IPARS framework sets up the subdomains for each processor. At this point, depending on the physical model, the IPARS framework performs the model initialization. Initial saturation and pressure are set here. After the initialization the IPARS framework gives the control to the main physical model. The main physical routine for the dual porosity models is the `DSTEP` subroutine.
The IPARS framework consists of three layers:

- Executive layer which provides an interface for directing the overall simulation process.
- Work routine layer which consists of FORTRAN subroutines. These routines are the engine of the simulator. These routines perform the gridblock calculations.
- Data management layer that handles the distribution of grid across processors, the local memory allocation, communication scheduling, and dynamic load balancing.

More details of the major executive and work routines in the compositional and chemical dual porosity models as well as the full tensor routines are covered in the following sections.

3.3 Executive Routines

The IPARS framework provides several interfaces for implementing any physical model. These interfaces have the same structure for all physical models. They provide several routines for invoking a physical model. The main executive routines in the dual porosity models are as follows:

`IPARS`	The main driver of the simulator. This is the main routine of the code. All models including dual porosity models are called from this routine.
`SETPRCS`	Sets multiprocessor parameters including number of processors, processor number, and process ID (if appropriate).
`READER`	Reads the data up to a terminator into a single character string for later analysis.
`XISDAT`	Reads initial scalar data for the fracture system. Scalar data include the component physical properties, default value for the number of the iteration, convergence tolerances, output flags, operation specific flags, number of phases, and chemical property data. Grid-element array are not allowed in this subroutine.
`XARRAY`	Allocates memory for all grid-element arrays for the fracture system.
`XARYDAT`	This is a header file which creates a common block for all pointers in the subroutine `XARRAY`.
`DARRY`	Allocates memory for all grid-element arrays for the matrix system.
`MATRIX`	This is a header file which creates a common block for storing all scalar variables of dual porosity models.
`XIADAT`	This subroutine reads the entire grid-element arrays such as pressure, water saturation, water viscosity, and compositions for the fracture system from the input file.

`DIADAT`	This subroutine reads the entire grid-element arrays such as pressure, water saturation, water viscosity, and compositions for the matrix system from the input file.
`XIVDAT`	Performs a complete model initialization for the fracture system. The PETSc linear solver is also initialized by this subroutine.
`DIVDAT`	Performs a complete model initialization for the matrix system.
`DSTEP`	The main interface between IPARS framework and compositional simulators. All calculations over a time-step for both fracture and matrix systems are performed by this subroutine.
`DPRINT`	Dumps out the solutions for both the fracture and matrix systems.
`XQUIT`	Stops simulation if the maximum time is achieved or if an error occurs. This subroutine finalizes the PETSc linear solver.

The main subroutine for the dual porosity models is `DSTEP`. This is the interface between the IPARS framework and compositional models. This subroutine is called by IPARS without any argument. The grid-element variables and dimensions are passed into this subroutine through pointers that are stored in common blocks. The grid-element arrays and dimensions are passed to this subroutine through a C routine called `CALLWORK`, which is handled by IPARS. Figure 3.1 shows a flowchart of GPAS routines. The `DSTEP` routine is called at each timestep. It performs the following tasks:

1. Initializes the indexing for the linear solver.

2. Calculates the dependent variables from the independent variables for both the fracture and matrix systems.

3. Updates the fracture properties in the ghost layers of all processors.

4. Calculates the residuals of the governing equations for both the fracture and matrix systems.
5. Selects the maximum residual for each governing equation among all processors in both the fracture and matrix media.
6. Checks the convergence criteria for both the fracture and matrix media. The check for the convergence is done in the master processor and broadcasts among all processors.
7. If the convergence is met in both the fracture and matrix media, it outputs the results at the current time level and then returns to the framework for the next timestep.
8. If the convergence failed on either of the fracture or matrix system, it constructs the Jacobian matrix for the fracture system.
9. Calculates the Jacobian matrix for the matrix system and modifies the elements of the Jacobian matrix of the fracture system using the Schur complement method (see Section 4.3 on page 72).
10. Solves the fracture system and calculates the fracture independent variables.
11. Determines the matrix independent variables using the fracture solutions.
12. Updates the dependent variables for both the fracture and matrix media and returns to step 2 for the next Newton iteration if required.

3.4 Work Routines

The main work routines implemented in the dual porosity models are as follows:

`TRANC1`	Calculates transmissability constant array for orthogonal grid option
`TRANC2`	Calculates transmissability constant array for corner-point grid option. This subroutine supports the full transmissibilty tensor calculation
`INFLUID0`	This subroutine performs the initialization and calculation of the initial fluid in place in the fracture system. This subroutine is called by IPARS framework at the beginning of the simulation
`INFLUIDM0`	This subroutine performs the initialization and calculation of the initial fluid in place in the matrix system. It also calculates the total fluid in place
`INFLUID`	This subroutine computes the fluid in place in the fracture system at both reservoir and standard conditions
`INFLUIDM`	This subroutine computes the fluid in place in the matrix system as well as total fluid in place at both reservoir and standard conditions
`PROP`	Computes the rock and fluid properties in the fracture system
`PROPM`	Computes the rock and fluid properties in the matrix system
`RESIDUAL`	Computes the residuals of the fracture equations
`RESIDUALM`	Computes the residuals of the matrix equations
`IWELL`	Inputs both initial and transient well data and calculates the locations of the well elements on the current processor
`JSOURCE`	Computes the derivatives of the source terms with respect to the fracture primary variables
`JACCUM`	Computes the derivatives of the accumulation term of the fracture system with respect to the fracture primary variables

`DJACCUM`	Computes the derivatives of the accumulation term of the matrix system with respect to the matrix primary variables
`JXMASS`	Computes the derivatives of the residuals of the mass balance equations in the fracture system with respect to fracture primary variables in the x direction
`JXMASSG`	Computes the derivatives of the residuals of the mass balance equations in the fracture system with respect to fracture primary variables in the x direction for a corner point mesh
`JYMASS`	Computes the derivatives of the residuals of the mass balance equations in the fracture system with respect to fracture primary variables in the y direction
`JYMASSG`	Computes the derivatives of the residuals of the mass balance equations in the fracture system with respect to fracture primary variables in the y direction for a corner point mesh
`JZMASS`	Computes the derivatives of the residuals of the mass balance equations in the fracture system with respect to fracture primary variables in the z direction
`JZMASSG`	Computes the derivatives of the residuals of the mass balance equations in the fracture system with respect to fracture primary variables in the z direction for a corner point mesh
`JHMASS`	Computes the derivatives of the residual of the mass balance equations in the matrix system with respect to the matrix primary variables in the horizontal direction
`JVMASS`	Computes the derivatives of the residual of the mass balance equations in the matrix system with respect to the matrix primary variables in the vertical direction

`JMFTH`	Computes the derivatives of the matrix-fracture transfer function in the horizontal direction with respect to the fracture and matrix primary variables
`JMFTV`	Computes the derivatives of the matrix-fracture transfer function in the vertical direction with respect to the fracture and matrix primary variables
`JACOBIAN`	Constructs the Jacobian matrix for the fracture system
`DJACOBIAN`	Constructs the Jacobian matrix for the matrix system within each fracture gridblock and then solves the system and modifies the Jacobian elements of the fracture system
`SOLVERS`	Solves the linear system of equations for the fracture system (Eq. 4.89)
`DSOLVER`	Solves the linear system of equations for the matrix system (Eqs. 4.92 and 4.93)
`DUPDATE`	Computes the matrix primary variables from the fracture solutions (Eq. 4.87)
`XDELTA`	Updates the dependent variables of the fracture system
`DDELTA`	Updates the dependent variables of the matrix system

3.5 PETSc Linear Solver

The Portable, Extensible Toolkit for Scientific Computations (PETSc) is a large collection of parallel, general-purpose, object-oriented, linear and nonlinear solvers for the scalable solution of partial differential equations (PDE) discretized using implicit and semi-implicit methods (Balay *et al.*, 1998; Wang *et al.*, 1999). PETSc is implemented in C and has interfaces for C, C++, and FORTRAN. It employs the MPI standard for all message-passing communication. The PETSc toolkit provides a rich environment to ease the development of scalable PDE-based simulation codes. It has been widely used in many computational applications, including nano-simulations, geosciences, environmental and subsurface flow simulations, computational fluid dynamics, wave propagation, structural dynamics, and econometrics.

Several Krylov methods are provided by the linear solver components of PETSc methods such as conjugate gradient (CG), generalized minimal residual (GEMRES) and biconjugate gradient (BCG). PETSc also provides several parallel preconditioners such as Jacobi, block preconditioners like block Jacobi, and domain decomposition preconditioners like additive Schwatrz.

Krylov subspace methods (KSP) and preconditioners are the essential tool for solving sparse linear systems of the form $Ax = b$. GPAS uses the linear solver component (KSP) of PETSc to solve the linearized Newton system of equations and uses the parallel data formats provided by PETSc to store the Jacobian matrix and the vectors. GPAS uses the Krylov method and a block Jacobi preconditioner, with point block incomplete factorization (ILU) on the subdomain blocks. The point block means a single gridblock is treated as a single unit. The number of the blocks for the block Jacobi is equal to the number of the processors. By which, each processor gets a complete subdomain of the problem and does a single local incomplete factorization on the Jacobian matrix corresponding to this subdomain.

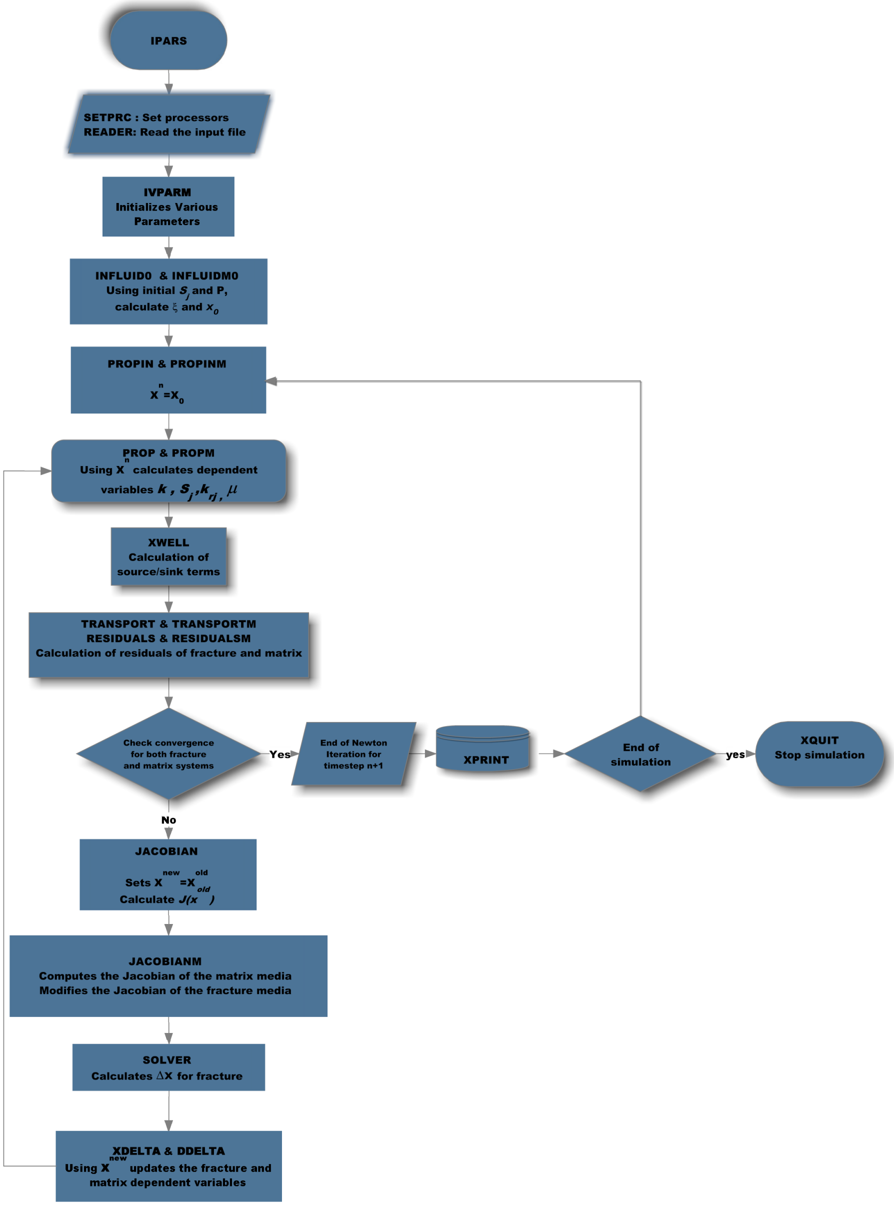


Figure 3.1: Flowchart of GPAS routines for dual porosity modules

Chapter 4

Compositional and Chemical Models for Fractured Reservoirs

This chapter covers the assumption, derivation, and formulation used in both equation of state (EOS) compositional and chemical dual porosity models in GPAS.

4.1 EOS Compositional Model Description

A fully implicit EOS compositional model was implemented in GPAS (Wang *et al.*, 1997). A brief review of assumptions and formulation of governing equations and phase equilibrium calculation will be covered in this section.

4.1.1 Assumptions

Development and formulation of the governing equations in GPAS are based on the following assumptions:

1. Darcy's law describes the multiphase flow in both matrix and fracture systems
2. Isothermal reservoir

3. The aqueous phase consists of only the water component
4. There is no mass transfer between the aqueous and oleic phases
5. The rock is slightly compressible and immobile
6. Water is slightly compressible and water viscosity is constant
7. Dispersion term is not included in this implementation
8. Matrix blocks are partially or totally surrounded by fractures or impermeable boundaries
9. The injection and production of fluids are treated as source or sink terms and wells are connected only to the fracture system
10. The fracture system has finite transmissibility and is under vertical equilibrium conditions, *i.e.*, capillary pressures in the fracture system are zero

4.1.2 Governing Equations

In dual porosity models, the reservoir grid is divided into two partitions, one for fracture which has all fracture properties such as permeability and porosity and another one for matrix. Hence, at each point in the system there are two values for each parameter.

The important transport mechanisms occurring in porous media are viscous forces, gravity forces, dispersion (diffusion), and capillary forces (Lake, 1989). The equation of motion and component mass conservation are written for each system independently and should hold at any point in the system at all times. The exchange of fluid between two systems is taken into account by a transfer function. Isothermal multicomponent and multiphase flow in porous media can be described by i) component material balance equations ii) volume constraint equations, and iii) phase equilibrium equations.

4.1.2.1 Material Balance Equations

The general mass conservation equation for species i in a volume V can be expressed as

$$\begin{aligned}\{\text{Rate of accumulation of } i \text{ in } V\} &= \{\text{Rate of } i \text{ transported into } V\} \\ &- \{\text{Rate of } i \text{ transported from } V\} \\ &+ \{\text{Rate of production of } i \text{ in } V\} \end{aligned} \quad (4.1)$$

$$i = 1, \ldots, n_c$$

The differential form of conservation equation for any species i can be expressed as (Lake, 1989)

$$\frac{\partial W_i}{\partial t} + \vec{\nabla} \cdot \vec{N}_i - R_i = 0 \quad (4.2)$$

where W_i is the overall concentration of i in units of mass of i per unit bulk volume, N_i is the flux vector of species i in units of mass of i per surface area-time, and R_i is the mass rate of production in units of mass of i per bulk volume-time.

The mass balance equation (Eq. 4.2) can be expressed in terms of moles per unit time by defining each term of Eq. 4.2 in terms of the porous media and fluid properties. The overall concentration can be expressed by

$$W_i = \phi \sum_{j=1}^{n_p} \xi_j S_j x_{ij} \quad (4.3)$$

where ϕ is the porosity, ξ_j is the molar density of phase j, S_j is the saturation of phase j, and x_{ij} is the mole fraction of component i in phase j. The flux vector of component i is a sum of the convective and the dispersive flux, and can be expressed as

$$\overrightarrow{N_i} = \sum_{j=1}^{n_p} \xi_j x_{ij} \overrightarrow{u}_j + \phi \xi_j S_j \overrightarrow{\overrightarrow{K}}_{ij} \cdot \overrightarrow{\nabla} x_{ij} \tag{4.4}$$

where $\overrightarrow{u}_j$ represents the superficial velocity or flux of phase j . The flux is evaluated using the Darcy's law for multiphase flow of fluids through porous media:

$$\overrightarrow{u}_j = -\bar{\bar{k}} \lambda_{rj} \left(\nabla P - \gamma \nabla D \right) \tag{4.5}$$

Darcy's law is a fundamental relationship describing the flow of fluids in permeable media under laminar flow conditions. The differential form of Darcy's law can be used to treat multiphase unsteady state flow, non-uniform permeability, and non-uniform pressure gradients. It is used to govern the transport of phases from one cell to another under the local pressure gradient, rock permeability, relative permeability and viscosity. Converting each of the terms in the mass balance equation to units of moles per unit time and expressing the flux using Darcy's law, the mass balance for each component i for fracture and matrix systems can be expressed by the following partial differential equation:

Fracture system:

$$\begin{aligned} & V_{fb} \frac{\partial}{\partial t} \left(\phi_f N_{fi} \right) - \\ & V_{fb} \overrightarrow{\nabla} \cdot \left[\sum_{j=1}^{n_p} \xi_{fj} \bar{\bar{k}} \lambda_{fj} x_{fij} \cdot \left(\overrightarrow{\nabla} P_{fj} - \gamma_{fj} \overrightarrow{\nabla} D_f \right) \right] \\ & - q_i + \tau_{mfi} = 0 \end{aligned} \tag{4.6}$$

Matrix system:

$$\begin{aligned} & V_{mb}\frac{\partial}{\partial t}\left(\phi_m N_{mi}\right) - \\ & V_{mb}\overrightarrow{\nabla}\cdot\left[\sum_{j=1}^{n_p}\xi_{mj}\bar{\bar{k}}\lambda_{mj}x_{mij}\cdot\left(\overrightarrow{\nabla}P_{mj}-\gamma_{mj}\overrightarrow{\nabla}D_m\right)\right]=0 \end{aligned} \tag{4.7}$$

for $i = 1, 2, \ldots, n_c$

In the above equations

$$\lambda_j = \frac{k_{rj}}{\mu_j} \tag{4.8}$$

and

$$\gamma_j = \frac{g}{g_c}\rho_j \tag{4.9}$$

where

N_i = Moles of component i per unit pore volume

x_{ij} = Mole fraction of component i in phase j

ξ_j = Molar density of phase j $\frac{lbmol}{ft^3}$

λ_j = Relative mobility of phase j, $\frac{1}{cp}$

γ_j = Specific gravity of phase j, $\frac{psia}{ft}$

ρ_j = Density of phase j, $\frac{lbmol}{ft^3}$

q_i = Volumetric molar injection or production rate , $\frac{lbmol}{d}$

τ_{mfi} = Matrix-fracture transfer function for component i, $\frac{lbmol}{d-ft^3}$

S_j = Saturation of phase j, fraction

D = Depth, ft

P_j = Pressure of phase j, $psia$

n_p = Total number of phases

n_c = Total number of hydrocarbon components

g_c = Gravitational conversion factor, $\frac{lb_m - ft}{lb_f - s^2}$

$\overrightarrow{\overrightarrow{K}}_{ij}$ = Dispersion coefficient tensor of component i in phase j, $\frac{ft^2}{d}$

ϕ = Porosity, fraction

Considering $x_{ij} = 1$ for water phase the conservation equations can be written as follow:

Fracture system:

$$\begin{aligned} & V_{fb} \frac{\partial}{\partial t} \left(\phi_f N_{fw} \right) - \\ & V_{fb} \overrightarrow{\nabla} \cdot \left[\xi_{fw} \bar{\bar{k}} \lambda_{fw} \cdot \left(\overrightarrow{\nabla} P_{fw} - \gamma_{fw} \overrightarrow{\nabla} D_f \right) \right] - q_w + \tau_{mfw} = 0 \end{aligned} \tag{4.10}$$

Matrix system:

$$\begin{aligned} & V_{mb} \frac{\partial}{\partial t} \left(\phi_m N_{mw} \right) - \\ & V_{mb} \overrightarrow{\nabla} \cdot \left[\xi_{mw} \bar{\bar{k}} \lambda_{mw} \cdot \left(\overrightarrow{\nabla} P_{mw} - \gamma_{mw} \overrightarrow{\nabla} D_m \right) \right] = 0 \end{aligned} \tag{4.11}$$

Based on the definition of dual porosity models, no source or sink terms are allowed in the matrix equations. The wells are connected to the fracture system.

4.1.2.2 Phase Equilibrium Equations

One of the criteria for phase equilibrium is the equality of the partial molar Gibbs free energies or the chemical potentials. Alternatively, this criterion can be expressed in terms of fugacity (Sandler, 1999).

With the assumption of local thermodynamic equilibrium for the hydrocarbon phases, the criterion of phase equilibrium applies (Smith and Van Ness, 1975):

Fracture system:

$$\ln(f_{fi}^g) - \ln(f_{fi}^o) = 0 \tag{4.12}$$

Matrix system:

$$\ln(f_{mi}^{g}) - \ln(f_{mi}^{o}) = 0 \tag{4.13}$$

In the above equations $f_i^j = \ln(x_{ij}\phi_{ij})$, where ϕ_{ij} is the fugacity coefficient of componet i in phase j.

4.1.2.3 Volume Constraint Equations

The volume constraint states that the pore volume in each cell must be filled completely by the total fluid volume. The volume constraints for fracture and matrix systems in mathematical forms can be expressed as
Fracure system:

$$\sum_{i=1}^{n_c+1} N_{fi} \sum_{j=1}^{n_p} L_j \bar{\nu}_j - V_{pf} = 0 \tag{4.14}$$

Matrix system:

$$\sum_{i=1}^{n_c+1} N_{mi} \sum_{j=1}^{n_p} L_j \bar{\nu}_j - V_{pm} = 0 \tag{4.15}$$

where N_i is the number of moles of component i per unit pore volume, L_j is the ratio of moles in phase j to the total number of moles on the mixture, V_p is the pore volume, and $\bar{\nu}_j$ is the specific molar volume of phase j.

4.1.3 Transfer Functions

The transfer functions for each component can be expressed by the following equations:

For hydrocarbon components:

$$\tau_{mfi} = NM \cdot V_b \sum_{l=1}^{N_b} \frac{\partial}{\partial t} (\phi_m N_{mi})_l \tag{4.16}$$

For water component:

$$\tau_{mfw} = NM \cdot V_b \sum_{l=1}^{N_b} \frac{\partial}{\partial t} (\phi_m N_{mw})_l \tag{4.17}$$

where NM is the total number of matrix blocks in a fracture gridblock and N_b is the number of matrix subgrids. The boundary condition for the matrix system is the continuity of all phase pressures and for the fracture system is the no-flow boundary condition.

4.1.4 Choice of Independent Variables and Solution Procedure

Equations 4.6 through 4.15 are the governing equations describing the fluid flow through porous media in naturally fractured reservoirs. The total number of equations is $2n_c + 2$. The independent unknown variables are chosen as $\ln K_i, N_i, P_w$, and N_w (N is moles per unit pore volume) for each system, fracture and matrix systems, which includes $2(n_c + 2)$ independent variables. This set is likely to be the best choice mainly because the fugacity equations become more linear with this set (Wang *et al.*, 1999).

The fluid physical properties and variables used in governing equations are expressed as a function of these independent variables. The governing equations further discretized using finite different approximations and are linearized using a Newton's procedure. The derivatives of the discretized governing equations with respect to independent variables are calculated from the Jacobian matrix. The linear system is further reduced using the phase equilibrium relationships for both fracture and matrix systems.

The Schur complement method is used to eliminate the matrix system unknowns and decouple matrix equations from the fracture equations so that the solution time does not become an encumbrance as the number of matrix subdomains increases. The linear system is solved for pressure and overall moles of components per unit bulk volume for the fracture system. The matrix unknowns are then back calculated using the fracture variables. The following steps explain the solution procedure in more detail (Naimi-Tajdar *et al.*, 2007).

1. Initializing fluid properties and primary variables for both matrix and fracture systems.
2. Performing phase stability test to calculate the number of phases and phase state. If necessary, performing flash calculation to determine phase density and compositions for each cell. Labeling all phases as oil, gas, or aqueous phase. Determining viscosity and relative permeability for each phase.
3. Linearizing both fracture and matrix governing equations and forming the Jacobian matrix.
4. Decoupling the matrix equations from the fracture equations and reducing the linear system using Schur complement method.
5. Solving the reduced linear system to obtain fracture independent variables.
6. Solving the decoupled matrix equations to obtain the matrix independent variables.
7. Updating the physical properties for both fracture and matrix systems.
8. Checking the convergence of both fracture and matrix systems. The convergence criteria are based on the residual tolerances for both fracture and matrix systems. If the convergence criteria are met, a new time-step is then started

by returning to step 3. Otherwise, another Newton iteration will be performed by returning to step 4.

4.1.5 Fluid-Related Calculation Using EOS

The phase equilibrium calculation is the heart of a compositional simulator. Equations of state, such as Soave (1972) and Peng and Robinson (1976) are increasingly used in reservoir simulators to determine thermodynamical properties. The number, amount, and compositions of phases are determined using the phase equilibrium calculation.

4.1.5.1 Phase Behavior

The Peng Robinson equation of state (PR-EOS) which is used in GPAS is expressed as

$$P = \frac{RT}{v-b} - \frac{a(T)}{v\,(v+b) + b\,(v-b)} \tag{4.18}$$

The constants $a(T)$ and b for a pure component are computed from

$$\begin{aligned} a(T) &= \Omega_a \frac{\alpha (RT_c)^2}{P_c} \\ b &= \Omega_b \frac{RT_c}{P_c} \end{aligned} \tag{4.19}$$

with

$$\begin{aligned} \Omega_a &= 0.45724 \\ \Omega_b &= 0.07780 \\ \alpha &= \left[1 + m\left(1 - \sqrt{\frac{T}{T_c}}\right)\right]^2 \end{aligned} \tag{4.20}$$

where

$$
\begin{aligned}
& m = 0.37464 + 1.54226\omega - 0.26992\omega^2 \\
& \text{if } \omega < 0.49 \\
& m = 0.379640 + 1.485030\omega - 0.164423\omega^2 + 0.016666\omega^3 \\
& \text{if } \omega \geq 0.49
\end{aligned}
\tag{4.21}
$$

When applied to multicomponent mixtures, the mixing rules are given by

$$
\begin{aligned}
& a = \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} x_i x_j \sqrt{a_i a_j} \left(1 - \delta_{ij}\right) \\
& b = \sum_{i=1}^{n_c} x_i b_i
\end{aligned}
\tag{4.22}
$$

where for each component, a_i and b_i are computed from Eq. 4.19.

The PR EOS can also be expressed in the compressibility factor, Z-factor, as

$$
Z^3 - (1 - B)Z^2 + (A - 3B^2 - 2B)Z - (AB - B^2 - B^3) = 0 \tag{4.23}
$$

where

$$
\begin{aligned}
& Z = \frac{P\nu}{RT} \\
& A = \frac{aP}{(RT)^2} \\
& B = \frac{bP}{RT}
\end{aligned}
\tag{4.24}
$$

4.1.5.2 Phase Stability Analysis

The stability of a phase is determined by the Gibbs free energy. A system is said to be in equilibrium if there are no unbalanced forces within the system. A phase is said to be stable if and only if among all admissible states at a constant pressure and temperature, the Gibbs free energy be at a minimum. The stability analysis is a search for a trial phase taken from the overall mixture composition $\bar{z}$ to form a state that minimizes the Gibbs free energy. If such a search is successful, an additional phase must be considered (Chang, 1990). Mathematically it can be expressed by

$$\Delta G = \sum_{i=1}^{n_c} y_i \left[\mu_i(\bar{y}) - \mu_i(\bar{z})\right] \tag{4.25}$$

where μ_i is the chemical potential of component i, $\bar{y}$ is the phase composition of the trial phase and y_i is the mole fraction of component i in the trial phase. A value of ΔG greater than zero indicates stability.

4.1.5.3 Flash Claculation

If the stability test demonstrates a two phase system, the compositions and amount of each phase are determined by a flash calculation. At first, the Rachford-Rice equation is solved by a Newton's method.

$$r(\nu) = \sum_{i=1}^{n_c} \frac{(K_i - 1)Z_i}{1 + \nu(K_i - 1)} \tag{4.26}$$

where ν is the mole fraction of gas in the absence of water, K_i is the equilibrium ratio, Z_i is the overall mole fraction of component i in the mixture, and $r(\nu)$ is the residual of the Rachford-Rice equation. The compositions of liquid and gas are then calculated from the following equations:

$$\begin{aligned} x_i &= \frac{Z_i}{1+\nu(K_i-1)} \\ y_i &= \frac{Z_i K_i}{1+\nu(K_i-1)} \end{aligned} \tag{4.27}$$

x_i and y_i should also satisfy the following criteria:

$$\begin{aligned} &\sum_{i=1}^{n_c} x_i - 1 = 0 \\ &\sum_{i=1}^{n_c} y_i - 1 = 0 \end{aligned} \tag{4.28}$$

4.1.5.4 Phase Identification and Tracking

All phases must be identified as oil, gas, or aqueous phase at the initial condition and also when a new phase appears. After a phase has been identified, phase tracking performs the labeling of the phase during simulation. Calculation of physical properties are based on the phase identification. Perschke (1988) developed a procedure for the phase identification and tracking using both phase mass density and phase compositions. This procedure has been implemented in GPAS.

When the hydrocarbon mixture is a single phase, an algorithm similar to that proposed by Gosset *et al.* (1986) is used. The parameters A and B of the Peng Robinson EOS are computed from

$$A = \frac{aP}{(RT)^2} = \Omega_a \frac{RT_c^2}{P_c T^2}\alpha \tag{4.29}$$

$$B = \frac{bP}{RT} = \Omega_b \frac{PT_c}{P_c T} \tag{4.30}$$

Dividing Eq. 4.29 by Eq. 4.30 gives

$$\frac{A}{B} = \frac{\Omega_a T_c}{\Omega_b T}\alpha \tag{4.31}$$

where Ω_a, Ω_b, and α are defined in Eq. 4.20. A fluid is said to be single phase if $T > T_c$, which also implies that $\alpha \leq 1$. From Eq. 4.31, this implies that

$$\frac{A}{B} \leq \frac{\Omega_a}{\Omega_b} \tag{4.32}$$

or its molar volume to be greater than the critical molar volume, $\nu > \nu_c$, which also implies that

$$Z > \frac{BZ_c}{\Omega_b} \tag{4.33}$$

Another option is also provided in the code to identify a single phase by conventional method. The fluid is liquid when $\sum_{i=1}^{n_c} Z_i K_i = 1$, and the fluid is gas when $\sum_{i=1}^{n_c} \frac{Z_i}{K_i} = 1$.

4.1.6 Physical Property Models

In this section, the physical models implemented in GPAS to calculate the viscosities, interfacial tension, relative permeability, and capillary pressure are described.

4.1.6.1 Viscosity

The aqueous phase viscosity is entered as a constant in the input file. The Lohrenz *et al.* (1964) correlation is used to determine the oil and gas viscosities. An option is also provided to input oil viscosity as a constant. The Lohrenz *et al.* (1964) correlation combines several viscosity correlations. The following steps are taken in calculation of the phase viscosity:

- Computation of the low-pressure, pure-component viscosity:

$$\begin{aligned} \widehat{\mu}_i &= \frac{0.00034T_{ri}^{0.94}}{\varsigma_i} \qquad \text{for } T_{ri} \le 1.5, \text{ or} \\ \widehat{\mu}_i &= \frac{0.0001776(4.58T_{ri} - 1.67)^{5/8}}{\varsigma_i} \qquad \text{for } T_{ri} > 1.5 \end{aligned} \tag{4.34}$$

where

$$\varsigma_i = \frac{5.44T_{ci}^{1/6}}{Mw_i^{1/2}P_{ci}^{2/3}} \tag{4.35}$$

where T_c is the critical temperature, P_c is the critical pressure, T_r is the reduced temperature, and Mw is the molecular weight.

- Calculation of low-pressure viscosity for mixture:

$$\mu_j^* = \frac{\sum\limits_{i=1}^{n_c} x_{ij}\widehat{\mu}_i\sqrt{Mw_i}}{\sum\limits_{i=1}^{n_c} x_{ij}\sqrt{Mw_i}} \tag{4.36}$$

- Calculation of the reduced phase molar density:

$$\eta_j = \frac{5.44\left[\sum\limits_{i=1}^{n_c} x_{ij}T_{ci}\right]^{1/6}}{\left[\sum\limits_{i=1}^{n_c} x_{ij}Mw_i\right]^{1/2}\left[\sum\limits_{i=1}^{n_c} x_{ij}P_{ci}\right]^{2/3}} \tag{4.37}$$

- Calculation of phase viscosity at desired pressure:

$$\begin{aligned} \mu_j &= \mu_j^* + 0.000205\frac{\xi_{jr}}{\eta_j} \qquad \text{for } \eta_{jr} \le 0.18 \\ \mu_j &= \frac{\mu_j^* + (\chi_j^4 - 1)}{10^4\eta_j} \qquad \text{for } \eta_{jr} > 0.18 \end{aligned} \tag{4.38}$$

where

$$\begin{aligned} \xi_{jr} &= \xi_j \sum_{i=1}^{n_c} x_{ij} V_{ci} \\ \chi_j &= 1.023 + 0.23364\xi_{jr} + 0.58533\xi_{jr}^2 - 0.40758\xi_{jr}^3 + 0.093324\xi_{jr}^4 \end{aligned} \tag{4.39}$$

4.1.6.2 Interfacial Tension

The interfacial tension between two hydrocarbon phases is determined using the Macleod-Sudgen correlation (Reid *et al.*, 1987).

$$\sigma_{jl} = \left[0.016018 \sum_{i=1}^{n_c} \psi_i (\xi_j x_{ij} - \xi_l x_{il}) \right]^4 \tag{4.40}$$

where ψ_i is the parachor of component i.

4.1.6.3 Relative Permeability

The two-phase relative permeabilty is given either as a tabular input or a Corey-type function. The expression for Corey model is expressed as

$$k_{rj} = k_{rj}^0 \bar{S}_j^{n_j} \tag{4.41}$$

where k_{rj} is the relative permeability of phase j, k_{rj}^0 is the endpoint relative permeability of phase j, n_j is the Corey model exponent, and $\bar{S}_j$ is the normalized phase saturation defined by

$$\bar{S}_j = \frac{S_j - S_{jr}}{1 - S_{j'r}} \tag{4.42}$$

In Eq. 4.42 S_j is the saturation of phase j, S_{jr} is the residual saturation of phase j, and $S_{j'r}$ is the residual saturation of conjugate phase of phase j.

Stone's model I or II is used to compute three-phase relative permeability of oil from two-phase relative permeability data.

- Normalized Stone I

  The Stone's Model I can be expressed as (Aziz and Settari, 1979):

$$k_{ro} = \frac{\bar{S}_o k_{row} k_{rog}}{k^0_{row}\left(1-\bar{S}_w\right)\left(1-\bar{S}_g\right)} \tag{4.43}$$

  where k_{row} is the two-phase water-oil relative permeability and k_{rog} is the two-phase oil-gas relative permeability. k_{row}, and k_{rog} are provided by input data. The normalized saturations are defined as

$$\begin{aligned}\bar{S}_o &= \frac{S_o - S_{or}}{1 - S_{or} - S_{wr}}\\ \bar{S}_g &= \frac{S_g}{1 - S_{or} - S_{wr}}\\ \bar{S}_w &= \frac{S_w - S_{wr}}{1 - S_{or} - S_{wr}}\end{aligned} \tag{4.44}$$

  Residual oil saturation during three-phase flow (S_{or}) is expressed as

$$S_{or} = bS_{orw} + (1-b)S_{org} \tag{4.45}$$

  Parameter b should define such that $b = 1$ when $S_g = 0$ and $b = 0$ when $S_w = S_{rw}$ (Fayers and Matthews, 1982). This implies that

$$b = 1 - \frac{S_g}{1 - S_{wr} - S_{rog}} \tag{4.46}$$

  In this model, it is assumed that $b = 0.5$, hence the residual oil saturation is a constant as

$$S_{or} = 0.5\left(S_{row} + S_{rog}\right) \tag{4.47}$$

- Normalized Stone II

  The Stone's model II is expressed as

  $$k_{ro} = k_{ro}^0 \left[\left(\frac{k_{row}}{k_{ro}^0} + k_{rw} \right) \left(\frac{k_{rog}}{k_{ro}^0} + k_{rg} \right) - (k_{rw} + k_{rg}) \right] \tag{4.48}$$

  The oil-water and oil-gas two-phase relative permeabilities are given in the input file.

4.1.6.4 Capillary Pressure

The capillary pressure data are entered in tabular format. The interpolation technique is used to determine capillary pressure at a given saturation. Capillary pressure data can be used for both fracture and matrix systems, even though in practice it is assumed that the fracture capillary pressure is zero.

4.2 Chemical Compositional Model Description

This section presents the formulation and assumptions used in the implementation of the chemical dual porosity model in GPAS. The first chemical model was implemented and validated for conservative tracers and polymers by Nalla (2002). This model was extended by John (2003) for modeling of partitioning tracer, chemical species adsorption, and calculating selected surfactant phase behavior. The key assumption in the hybrid method, developed by John (2003), is that chemical species such as tracers, polymer, surfactant and electrolytes occupy negligible volume and do not affect the EOS model governing equations, which is reasonable for typical chemical enhanced oil recovery processes and other applications such as aqueous tracers.

In the hybrid approach, the material conservation equations for hydrocarbon are solved implicitly while the aqueous species material balances are solved explicitly using updated phase fluxes, saturations, and densities. The disadvantages of the hybrid method are neglecting the amount of oil dissolved in aqueous phase and surfactant volume in volume constraint equation. Han *et al.* (2005) implemented a fully implicit chemical model in GPAS. The current implementation of chemical dual porosity model is based on Han's approach for the single porosity model.

4.2.1 Governing Equations

The current chemical model in GPAS is surfactant/oil/brine Type II(-) (Han *et al.*, 2005). There are up to only two phases under Type II(-) phase environment. The phases could be aqueous and oleic phases or microemulsion and oleic phases or a single microemulsion phase. The gas phase is not considered in the current chemical model in GPAS. The governing equations are i) material balance equations, and ii) volume constraints

4.2.1.1 Material Balance Equations

The oleic phase is assumed to be pure oil and microemulsion phase contains water, salt (electrolytes), polymer, surfactant, and solubilized oil. Hydrocarbon components can be dissolved in aqueous phase and there is no gas phase under Type II(-) surfactant phase environment. Therefore, material balance equations for the hydrocarbon component are expressed as

Fracture system:

$$\begin{aligned}
&V_{fb}\frac{\partial}{\partial t}\left(\phi_f N_{fi}\right)\\
&-V_{fb}\overrightarrow{\nabla}\cdot\left[\xi_{fi2}\frac{\bar{\bar{k}}_f k_{fr2}}{\mu_{f2}}x_{fi2}\cdot\left(\nabla P_f-\gamma_{f2}\nabla D_f\right)+\xi_{fi3}\frac{\bar{\bar{k}}_f k_{fr3}}{\mu_{f3}}x_{fi3}\cdot\left(\nabla P_f-\gamma_{f3}\nabla D_f\right)\right]\\
&-q_{fi}+\tau_{mfi}=0.0\\
&i=1,2,\ldots,n_c
\end{aligned}$$

$$(4.49)$$

Matrix system:

$$\begin{aligned} & V_{mb}\frac{\partial}{\partial t}\left(\phi_m N_{mi}\right) \\ & - V_{mb}\overrightarrow{\nabla}\cdot\left[\xi_{mi2}\frac{\bar{\bar{k}}_m k_{mr2}}{\mu_{m2}}x_{mi2}\cdot\left(\nabla P_m-\gamma_{m2}\nabla D_m\right)+\right. \\ & \qquad\qquad \left.\xi_{mi3}\frac{\bar{\bar{k}}_m k_{mr3}}{\mu_{m3}}x_{mi3}\cdot\left(\nabla P_m-\gamma_{m3}\nabla D_m\right)\right]=0.0 \\ & i=1,2,\ldots,n_c \end{aligned} \tag{4.50}$$

However, aqueous components such as water, surfactant, polymer, salt, and tracer are assumed to be dissolved in only aqueous phase. In this case, the material balance equations for the aqueous component are

Fracture system:

$$\begin{aligned} & V_{fb}\frac{\partial}{\partial t}\left(\phi_f N_{fi}\right) \\ & - V_{fb}\overrightarrow{\nabla}\cdot\left[\xi_{fi3}\frac{\bar{\bar{k}}_f k_{fr3}}{\mu_{f3}}x_{fi3}\cdot\left(\nabla P_f-\gamma_{f3}\nabla D_f\right)\right]-q_{fi}+\tau_{mfi}=0.0 \\ & i=n_c+1,\ldots,n_c+n_a+1 \end{aligned} \tag{4.51}$$

Matrix system:

$$\begin{aligned} & V_{mb}\frac{\partial}{\partial t}\left(\phi_m N_{mi}\right) \\ & - V_{mb}\overrightarrow{\nabla}\cdot\left[\xi_{mi3}\frac{\bar{\bar{k}}_m k_{mr3}}{\mu_{m3}}x_{mi3}\cdot\left(\nabla P_m-\gamma_{m3}\nabla D_m\right)\right]=0.0 \\ & i=n_c+1,\ldots,n_c+n_a+1 \end{aligned} \tag{4.52}$$

where subscripts 2 refers to the oleic phase and 3 refers to the microemulsion phase.

n_c is the number of hydrocarbon components and n_a is the number of aqueous components excluding water. n_c+n_a+1 refers to the water component. If surfactant concentration is less than critical micelle concentration (CMC), then subscript 3 refers to aqueous phase only.

4.2.1.2 Volume Constraint Equations

As with the EOS compositional volume constraint, the sum of volumes of occupying components, which are surfactant, water, and hydrocarbon components, should be equal to the pore volume. In mathematical form it can be shown as

$$\frac{N_{H_2O}}{\xi_{H_2O}} + \frac{N_{surf}}{\xi_{surf}} + \frac{\sum\limits_{i=1}^{n_c} N_i}{\xi_o} = 1.0 \tag{4.53}$$

where

$$\begin{aligned} \xi_{H_2O} &= \xi_{H_2O}^o \left[1 + c_{H_2O}^o \left(P - P^o\right)\right] \\ \xi_{surf} &= \xi_{surf}^o \left[1 + c_{surf}^o \left(P - P^o\right)\right] \end{aligned} \tag{4.54}$$

and ξ_o is calculated from the EOS compositional model.

Eq. 4.53 holds for both fracture and matrix systems using appropriate variables for each system.

4.2.2 Choice of Independent Variables and Solution Procedure

Equations 4.49 through 4.53 are solved simultaneously using the fully implicit finite difference scheme. After the finite differencing of Eqs. 4.49 through 4.53, there are $n_c + n_a + 2$ nonlinear equations with the same number of unknowns per each gridblock of the fracture and matrix. The independent unknown variables are chosen as N_i, P_w, and N_w. Since the gas phase is not considered in this implementation, there is no fugacity equation. The solution procedure is the same as the solution procedure of EOS compositional model, whereas the chemical phase behavior is used instead of flash calculation in Step 2 (for more detailes see Section 4.1.4 on page 49).

4.2.3 Fluid-Related Calculation

This section covers the effect of chemical species on the fluid properties. The surfactant/oil/brine phase behavior calculation and aqueous phase compositions will be presented.

4.2.3.1 Surfactant and Polymer Adsorption

Surfactant and polymer adsorptions are modeled as a function of concentrations and salinity using a Langmuir-type isotherm (Hirasaki and Pope, 1974).

$$\hat{C}_i = \min\left[\tilde{C}_i, \frac{a_i\left(\tilde{C}_i - \hat{C}_i\right)}{1 + b_i\left(\tilde{C}_i - \hat{C}_i\right)}\right] \tag{4.55}$$

where

$$a_i = a_{i1} + a_{i2}C_{SE} \tag{4.56}$$

and $\tilde{C}_i$ is the aqueous concentration of species i normalized by the water concentration C_1. Parameters a_{i1}, a_{i2}, and b_i are determined by matching surfactant ($i = 3$) or polymer ($i = 4$) adsorption data. Adsorption is assumed to be irreversible with respect to both salinity and concentrations. The minimum is taken to ensure that the adsorbed concentration is not greater than the total concentration.

4.2.3.2 Surfactant/Oil/Brine Phase Behavior

Certain molecules have two distinct components, differing in their affinity for solutes. The part of the molecule which has the tendency toward polar solute, such as water, is said to be hydrophilic. The part which has the tendency toward hydrocarbons is said to be hydrophobic. Amphiphilic molecules are molecules containing both types of components. An amphiphilic molecule can partition into the interface such that the polar part interacts with water and the non-polar part interacts with the hydrocarbon. Hence, it disrupts the cohesive energy at the surface and lowers the interfacial tension [Fig. 4.1(a)]. Such a molecule is called *surface active* or surfactant. Another form of arrangement is when the hydrophobic parts are oriented within the cluster and the hydrophilic parts are exposed to the solvent. This form is called micelle [Fig. 4.1(b)].

If the surfactant concentration is below the CMC, the oil does not get solubilized in the water and there are no changes in the interfacial tensions and relative permeabilities. The two phases are an aqueous phase containing all the surfactant, water, and electrolytes and a pure excess oil phase.

The surfactant/oil/brine phase behavior is based on Winsor (1954), Reed and Healy (1977), Nelson and Pope (1978), Prouvost *et al.* (1985), and Camilleri *et al.* (1987). In the absence of alcohol, three pseudo-components, (surfactant, oil, and brine) are used in the phase behavior calculations. The volumetric concentration of the three pseudo-components are used as the coordinates on a ternary diagram. The salinity, presence of alcohol, and divalent cations influence the phase behavior significantly, but in the current implementation it is assumed that the phase behavior is affected by effective salinity (C_{SE}) only. Depending upon the salinity and overall compositions, there are three types of phase environments.

A) At a sufficiently low salinity, an excess oil phase and a microemulsion phase exist. The excess oil phase is a pure oil and the microemulsion phase contains water, electrolytes, surfactants, and some solubilized oil. The plait point is close to or at the oil corner of the ternary diagram. There are two phases in this type and tie lines have negative slops. This type of phase behavior is called *Winsor Type I* or *Type II(-)* (Fig. 4.2).

B) As the effective salinity increases, the mixture separates into three phases at a critical salinity. The phases are an excess oil, water and a microemulsion phase (Fig. 4.3). This phase environment is called *Winsor Type III* or simply *Type III*. This critical effective salinity is called the lower effective salinity (C_{SEL}). At the lower effective salinity the invariant point is close to the brine pseudo-component. The invariant point moves toward the oil pseudo-component as the effective salinity increases. The three phase region exists until the effective salinity reaches another critical salinity called upper effective salinity (C_{SEU}). In this environment there exists a two-phase region as well.

C) At an effective salinity higher than the C_{SEU}, the surfactant partitions into oil and solubilizes water into an oil-external microemulsion phase. There are two phases in this environment; an excess water and a microemulsion phase containing most of the surfactant, oil, and some solubilized water. The plait point is close to or at the brine corner in the ternary diagram. The tie lines have positive slopes which name this type *Winsor Type II(+)* or simply *Winsor Type II* (Fig. 4.4).

The binodal curves, plait point, and tie lines are functions of effective salinity. Once they are modeled using a Hand's rule empirical correlation, then the surfactant/oil/brine phase behavior is expressed as a function of effective salinity. The phase behavior calculations are based on the following assumptions:

- Isothermal conditions
- Corner plait point formulation where the excess oil phase is a pure oil phase

- Binodal curve formulation based on Hand's rule (Hand, 1939)
- A symmetric binodal curve
- No effect of divalent cations such as calcium on the phase behavior
- No effect of alcohol on the phase behavior
- The phase behavior is independent of the polymer concentration
- No gas is present
- No effect of pressure on the microemulsion phase behavior

Based on empirical observations, the equilibrium phase concentration ratios can be represented by straight lines on a log-log scale as shown in Figure 4.5 (Hand, 1939). The binodal curve is given by the following equation:

$$\frac{C_{3j}}{C_{2j}} = A\left(\frac{C_{3j}}{C_{1j}}\right)^{B} \quad \text{for } j = 2, 3 \tag{4.57}$$

where A and B are defined empirically. C_{3j} and C_{2j} are the concentration of surfactant and oil in phase j. In C_{ij}, the first subscript (subscript i) represent the components (1=Water, 2=Oil, and 3=Surfactant) and the second subscript (subscripts j) represents the phases (1=Aqueous phase, 2=Oleic phase, and 3=Microemulsion phase). The parameter B is equal to -1 in a symmetric system. All component concentrations (C_{ij}) are volume fractions and must add up to one.

$$\sum_{i=1}^{3} C_{ij} = 1 \quad j = 1, 2, 3 \tag{4.58}$$

The parameter A is related to the height of binodal curve and is given by

$$A = (A_0 - A_1)\left(1 - \frac{C_{SE}}{C_{SEOP}}\right) + A_1 \quad \text{for } C_{SE} \leq C_{SEOP} \tag{4.59}$$

where C_{SE} is the salt concentration in aqueous phase and C_{SEOP} is the optimal salinity calculated from the lower and upper effective salinity as

$$C_{SEOP} = \frac{C_{SEL} + C_{SEU}}{2} \tag{4.60}$$

Parametes A_0 and A_1 are given as

$$A_s = \left(\frac{2C^s_{3max}}{1 - C^s_{3max}} \right)^2 \quad \text{for } s = 0, 1 \tag{4.61}$$

where $s = 0, 1$ represents the zero and optimal salinities.

4.2.3.3 Calculation of Surfactant Phase Composition

CMC is the key parameter in the calculation of surfactant phase composition.

A) Surfactant concentration below the CMC

In this case, the oil does not get solubilized in the water and there are no changes in the interfacial tensions and relative permeabilities. The two phases are an aqueous phase containing all the surfactant, water, electrolytes and a pure excess oil phase. The concentration of surfactant in the aqueous phase is calculated by

$$C_{31} = \frac{C_3}{C_1} \tag{4.62}$$

where C_1 and C_3 are the total concentrations (fluid volume fraction) of the brine and the surfactant.

B) Surfactant concentration above the CMC

For Type II(-) $C_{SE} \leq C_{SEL}$ and the surfactant solubilization ratio in the aqueous phase is given by

$$R_{31} = \frac{C_3^f}{C_1} \tag{4.63}$$

where $C_3^f = \tilde{C}_3 - \hat{C}_3$ is the concentration of surfactant in the fluid (total concentration minus adsorbed concentration) per pore volume. The solubilization ratio in oleic phase is given by:

$$R_{32} = \frac{A}{R_{31}} \tag{4.64}$$

where parameter A is given by Eq. 4.59. The microemulsion phase compositions are then determined as follows:

- For the water component

$$C_{13} = \frac{R_{32}}{R_{31} + R_{31}R_{32} + R_{32}} \tag{4.65}$$

- For the oil component

$$C_{23} = 1 - (1 + R_{31})\, C_{13} \tag{4.66}$$

- For the surfactant component

$$C_{33} = 1 - C_{13} - C_{23} \tag{4.67}$$

The oil solubilization ratio is expressed by

$$R_{23} = \frac{C_{23}}{C_{33}} \tag{4.68}$$

4.2.3.4 Interfacial Tension

The interfacial tension and solubilization ratio are related by the following function (Huh, 1979):

$$\sigma_{23} = \frac{c}{R_{23}^2} \tag{4.69}$$

Delshad *et al.* (1996) introduced Hirasaki (1981) correlation factor (Eq. 4.71) for modified Huh's equation so that the aqueous-oil interfacial tension reduces to the water-oil interfacial tension as the surfactant concentration approaches to zero.

$$\sigma_{23} = \sigma_{ow} e^{-aR_{23}} + \frac{cF_2}{R_{23}^2}\left(1 - e^{-aR_{23}^2}\right) \tag{4.70}$$

$$F_2 = \frac{1 - e^{-\sqrt{\sum\limits_{k=1}^{3}(C_{k2}-C_{k3})^2}}}{1 - e^{-\sqrt{2}}} \tag{4.71}$$

If $C_3 \leq CMC$ Eq. 4.70 reduces to

$$\sigma_{23} = \sigma_{ow} \tag{4.72}$$

4.2.3.5 Aqueous Phase Viscosity

At a fixed shear rate, the effect of polymer concentration on the aqeuous phase viscosity is expressed by Flory-Huggins equation (Flory, 1953).

$$\mu_p = \mu_w \left[1 + \left(A_{p1}C_{41} + A_{p2}C_{41}^2 + A_{p3}C_{41}^3\right) C_{SE}^{S_p}\right] \tag{4.73}$$

where $C_{41} = 100\frac{M_{poly}N^f_{poly}}{M_{H_2O}}$ and A_{p1}, A_{p2}, A_{p3}, and S_p are model parameters. The effect of surfactant on the viscosity is modeled by the following equation:

$$\mu_3 = C_{13}\mu_p e^{\alpha_1(C_{23}+C_{33})} + C_{23}\mu_o e^{\alpha_2(C_{13}+C_{33})} + C_{33}\alpha_3 e^{(\alpha_4 C_{13}+\alpha_5 C_{23})} \tag{4.74}$$

μ_w and μ_o are pure water and oil viscosities respectively.

4.2.3.6 Relative Permeability

The main goal of surfactant flooding is lowering the residual oil by reducing the interfacial tension between the reservoir oil and the injected water. Lowering the IFT results in eliminating capillary forces that are responsible for trapping one phase by another during an immiscible displacement in porous media and mobilizing the residual oil. This mobilization effect needs to be accurately captured in the relative permeability model. The trapping number is used in the relative permeability model to better represent the effect of lowering IFT.

A combination of viscous, buoyancy and capillary forces are incorporated in a dimensionless number called the trapping number (Jin, 1995; Pope *et al.*, 2000).

$$N_{Tl} = \frac{\left|\vec{\vec{k}} \cdot \vec{F}\right|}{\sigma_{ll'}} \tag{4.75}$$

where

N_{Tl} = The trapping number for phase l
$\vec{F}$ = The net driving force (viscous and capillary) on the trapped globule
$\vec{\vec{k}}$ = The permeability tensor
$\sigma_{ll'}$ = The IFT between displaced (l) and displacing (l') phases.

The net driving force is expressed by

$$\begin{aligned}
&\vec{F} = F_x\vec{i} + F_y\vec{j} + F_k\vec{k} \\
&F_x = -\frac{\partial \Phi_{l'}}{\partial x} \\
&F_y = -\frac{\partial \Phi_{l'}}{\partial y} \\
&F_z = -\frac{\partial \Phi_{l'}}{\partial z} - \left(\rho_{l'} - \rho_l\right) g
\end{aligned} \tag{4.76}$$

or in term of pressure

$$\begin{aligned}
&F_x = -\frac{\partial P_{l'}}{\partial x} \\
&F_y = -\frac{\partial P_{l'}}{\partial y} \\
&F_z = -\left(\frac{\partial P_{l'}}{\partial z} - \rho_{l'} g\right) - \left(\rho_{l'} - \rho_l\right) g = -\frac{\partial P_{l'}}{\partial z} + \rho_l g
\end{aligned} \tag{4.77}$$

The change in residual saturations is modeled as a function of trapping number as follow

$$S_{lr} = \min\left(S_l, S_{lr}^{High} + \frac{S_{lr}^{Low} - S_{lr}^{High}}{1 + T_l N_{tl}^{\tau}}\right) \tag{4.78}$$

where S_{lr}^{High} and S_{lr}^{Low} are the residual saturations for phase l at high and low trapping number. T_l and τ_l are calculated from the experimental data. This correlation was derived based on the experimental data for n-decane (Delshad, 1990).

The end points and exponents of the relative permeability curves are also functions of the trapping number (Morrow and Songkran, 1982; Morrow *et al.*, 1985; Delshad *et al.*, 1986). A linear interpolation method is used to estimate these changes from the given input data values at low and high trapping numbers.

$$k_{rl}^{o} = k_{rl}^{oLow} + \frac{S_{l'r}^{Low} - S_{l'r}}{S_{l'r}^{Low} - S_{l'r}^{High}} \left(k_{rl}^{oHigh} - k_{rl}^{oLow}\right) \tag{4.79}$$

$$n_l = n_l^{Low} + \frac{S_{l'r}^{Low} - S_{l'r}}{S_{l'r}^{Low} - S_{l'r}^{High}} \left(n_l^{High} - n_l^{Low} \right) \tag{4.80}$$

4.3 Solution Approach

The solution approach for both compositional EOS and compositional chemical models are the same. After discretizing and linearizing the governing equations (Equations 4.6 through 4.15 for EOS module and Eqs. 4.49 through 4.53 for chemical modules), the derivatives are calculated with respect to independent variables. In the matrix form the equations can be written as

$$-\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}} \right)^n \delta \mathbf{x}^{n+1} = \mathbf{R}^n \tag{4.81}$$

where $\mathbf{R}$ is the residual vector and n is the iteration level. The $\frac{\partial \mathbf{R}}{\partial \mathbf{x}}$ is called the Jacobian matrix which is the partial derivatives of all residual with respect to all unknowns.

$$\delta \mathbf{x}^{n+1} = \mathbf{x}^{n+1} - \mathbf{x}^n \tag{4.82}$$

where $\mathbf{x}$ is the vectors of unknowns (independent variables).

For a given gridblock i the fracture equations depend on the fracture unknowns of gridblock i, the fracture unknowns of gridblocks adjacent to gridblock i, and the matrix variables of gridblock i. The matrix equations only depend on the matrix variables and its surrounding fracture variables. The fracture and matrix residual equations can be expressed in the following form.

Fracture equations:

$$\bar{\bar{A}}_{ff} \mathbf{F} + \bar{\bar{B}}_{fm} \mathbf{M} = \mathbf{R}_f \tag{4.83}$$

Matrix equations:

$$\bar{\bar{C}}_{mf}\mathbf{F} + \bar{\bar{D}}_{mm}\mathbf{M} = \mathbf{R}_m \tag{4.84}$$

where

$\bar{\bar{A}}_{ff} = \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{x}_f}$ Jacobian of the fracture equations with respect to the fracture independent variables

$\bar{\bar{B}}_{fm} = \dfrac{\partial \mathbf{R}_f}{\partial \mathbf{x}_m}$ Jacobian of the fracture equations with respect to the matrix independent variables

$\bar{\bar{C}}_{mf} = \dfrac{\partial \mathbf{R}_m}{\partial \mathbf{x}_f}$ Jacobian of the matrix equations with respect to the fracture independent variables

$\bar{\bar{D}}_{mm} = \dfrac{\partial \mathbf{R}_m}{\partial \mathbf{x}_m}$ Jacobian of the matrix equations with respect to the matrix independent variables

$$\text{For EOS model } \mathbf{x} = \begin{bmatrix} \ln K_i \\ N_i \\ P_w \\ N_w \end{bmatrix} \text{ and for chemical model } \quad \mathbf{x} = \begin{bmatrix} N_i \\ P_w \\ N_w \end{bmatrix} \tag{4.85}$$

Parameters $\mathbf{F} = \delta \mathbf{x}_f$ and $\mathbf{M} = \delta \mathbf{x}_m$ are the fracture and matrix unknowns, respectively. $\mathbf{R}_f$ and $\mathbf{R}_m$ are the fracture and matrix residuals. In matrix form, Eq. 4.83 and Eq. 4.84 can be written as

$$\begin{pmatrix} \bar{\bar{A}}_{ff} & \bar{\bar{B}}_{fm} \\ \bar{\bar{C}}_{mf} & \bar{\bar{D}}_{mm} \end{pmatrix} \begin{pmatrix} \mathbf{F} \\ \mathbf{M} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_f \\ \mathbf{R}_m \end{pmatrix} \tag{4.86}$$

Solving for $\mathbf{M}$ using equation 4.84 yields:

$$\mathbf{M} = \bar{\bar{D}}_{mm}^{-1}\left(\mathbf{R}_m - \bar{\bar{C}}_{mf}\mathbf{F}\right) \tag{4.87}$$

Substituting $\mathbf{M}$ from Eq. 4.87 into Eq. 4.83 follows:

$$\bar{\bar{A}}_{ff}\mathbf{F} + \bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\left(\mathbf{R}_m - \bar{\bar{C}}_{mf}\mathbf{F}\right) = \mathbf{R}_f \tag{4.88}$$

Solving Eq. 4.88 for $\mathbf{F}$ gives

$$\left(\bar{\bar{A}}_{ff} - \bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\bar{\bar{C}}_{mf}\right)\mathbf{F} = \mathbf{R}_f - \bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\mathbf{R}_m \tag{4.89}$$

In Eq. 4.89 the only unknowns are the fracture unknowns ($\mathbf{F}$). The single porosity solver can be used to solve Eq. 4.89. Calculations of $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\bar{\bar{C}}_{mf}$ and $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\mathbf{R}_m$ are the big challenges in the solution of Eq. 4.89. In these calculations, the matrix inverse ($\bar{\bar{D}}_{mm}^{-1}$) is required. Computing the matrix inverse ($\bar{\bar{D}}_{mm}^{-1}$) is very expensive and not feasible. Instead of computing the matrix inverse the following procedure is used to calculate $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\bar{\bar{C}}_{mf}$ and $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\mathbf{R}_m$. Suppose

$$\mathbf{X}_m = \bar{\bar{D}}_{mm}^{-1}\mathbf{R}_m \tag{4.90}$$

and

$$\bar{\bar{\mathbf{Y}}}_{mf} = \bar{\bar{D}}_{mm}^{-1}\bar{\bar{C}}_{mf} \tag{4.91}$$

Then, solving the following systems of linear equations will give $\mathbf{X}_m$ and $\bar{\bar{\mathbf{Y}}}_{mf}$.

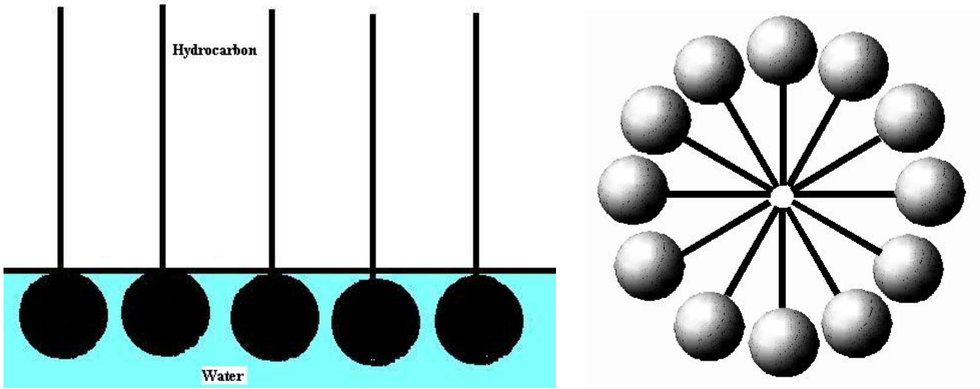
$$\bar{\bar{D}}_{mm}\mathbf{X}_m = \mathbf{R}_m \tag{4.92}$$

and

$$\bar{\bar{D}}_{mm}\bar{\bar{\mathbf{Y}}}_{mf} = \bar{\bar{C}}_{mf} \tag{4.93}$$

The Portable, Extensible Toolkit for Scientific Computation (PETSc) is used to solve Eqs. 4.92 and 4.93. The solution sequence for a dual porosity system using the above procedure is as follows:

1. Assemble the Jacobian matrices $\bar{\bar{D}}_{mm}, \bar{\bar{C}}_{mf}, \bar{\bar{B}}_{fm}$ and the residual vector $\mathbf{R}_m$ for the matrix system in each fracture.
2. Solve the system of linear equations, Eqs. 4.92 and 4.93, using PETSc to get $\mathbf{X}_m$ and $\bar{\bar{\mathbf{Y}}}_{mf}$. Solution of Eq. 4.92 is pretty straightforward, but the solution of Eq. 4.93 requires extra coding because the right-hand side of Eq. 4.93 is a matrix and PETSc does not have any routine for this kind of system.
3. Multiply $\bar{\bar{B}}_{fm}$ by $\mathbf{X}_m$ and $\bar{\bar{\mathbf{Y}}}_{mf}$ to get $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\bar{\bar{C}}_{mf}$ and $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\mathbf{R}_m$.
4. Modify the Jacobian matrix and residual vector of the fracture system by subtracting $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\bar{\bar{C}}_{mf}$ from $\bar{\bar{A}}_{ff}$ and $\bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\mathbf{R}_m$ from $\mathbf{R}_f$ to get the left and right-hand sides of Eq. 4.89.
5. Solve Eq. 4.89 using the single porosity solver to get the fracture unknowns $\mathbf{F}$. Eq. 4.89 is similar to $\mathcal{A}\mathbf{x} = \mathbf{b}$ where $\mathcal{A} = \bar{\bar{A}}_{ff} - \bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\bar{\bar{C}}_{mf}$ and $\mathbf{b} = \mathbf{R}_f - \bar{\bar{B}}_{fm}\bar{\bar{D}}_{mm}^{-1}\mathbf{R}_m$.
6. After calculating the fracture unknowns $\mathbf{F}$, Eq. 4.87 is used to calculate the matrix unknown $\mathbf{M}$.



(a) Surfactant at water-hydrocarbon interface (b) Micelle

Figure 4.1: Surfactant arrangements

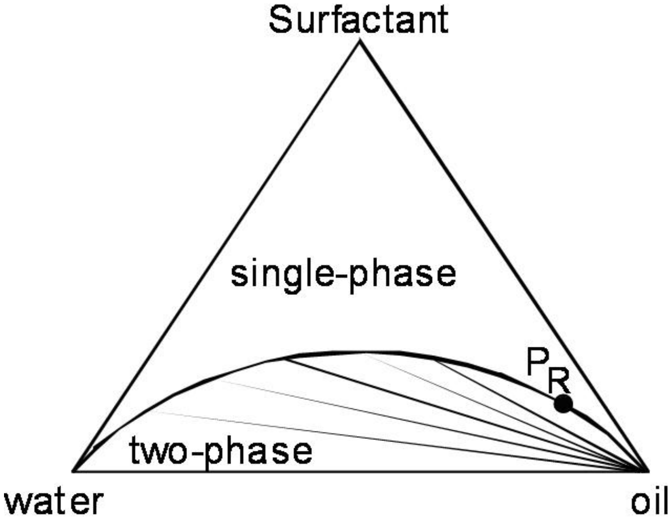


Figure 4.2: Type II(-) phase environment

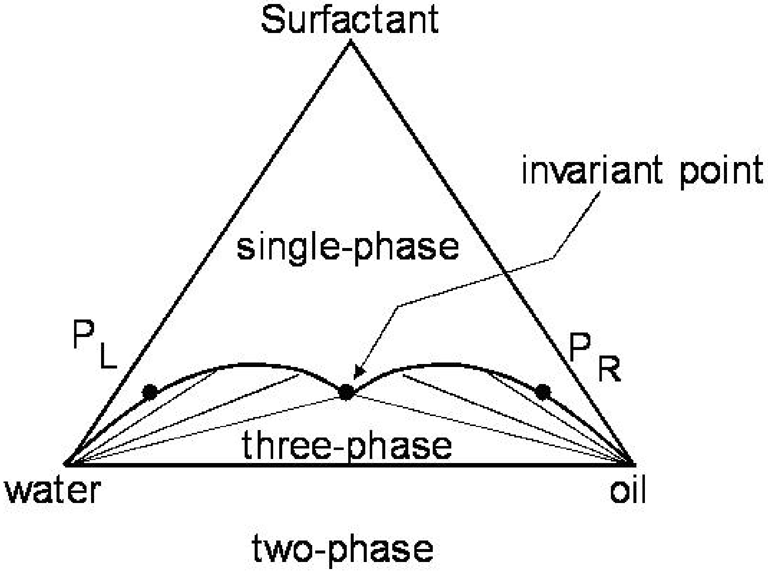


Figure 4.3: Type III phase environment

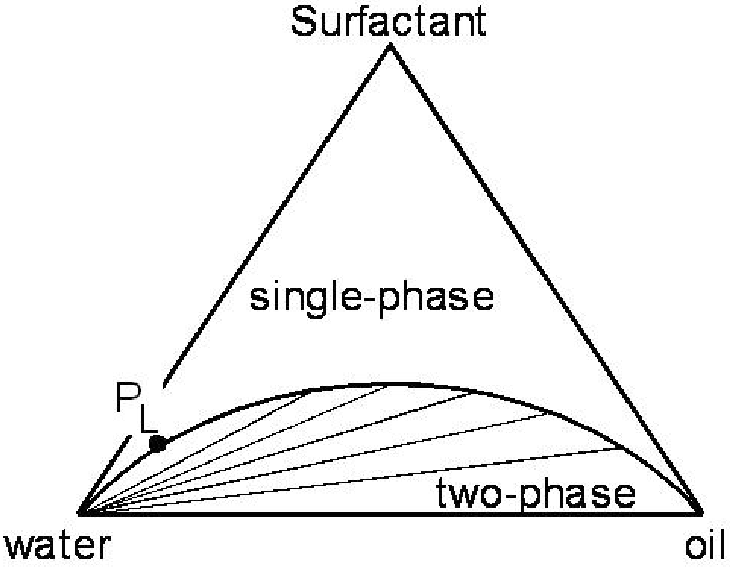
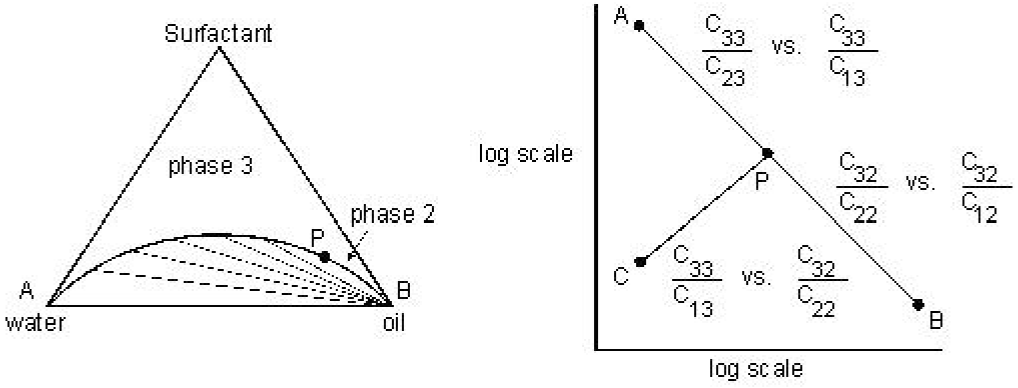


Figure 4.4: Type II(+) Phase Environment



(a) (b)

Figure 4.5: a) Ternary diagram and b) Hand plot for Type II(-)

Chapter 5

Computational Methods for Dual Porosity Models

The partial differential component material balance equations derived in Chapter 4 (Eqs. 4.6, 4.7, 4.10, and 4.11 for the EOS compositional model and Eqs. 4.49, 4.50, 4.51, and 4.52 for the compositional chemical model) need to be discretized using appropriate difference schemes in both space and time. An extensive discussion on the discretization of the material balance equations for both the EOS compositional model and the compositional chemical dual porosity models is given in this chapter. The solution procedure to solve the discretized form of the material balance equations together with the volume constraints and phase equilibrium equations will be covered in this chapter.

A fully implicit method is used to solve the governing equations. This method treats each term in Eq. 4.6 through Eq. 4.13 and Eq. 4.49 through Eq. 4.53 implicitly. These equations are nonlinear and need to be linearized using a Newton procedure. The Jacobian matrix, which is the matrix of all first-order partial derivatives of a vector-valued function, will be created. A suitable numerical method will be implemented to solve the systems of equations.

5.1 Numerical Formulation for EOS Model

This section covers the discretization and linearization of Eq. 4.6 through Eq. 4.11. The primary variables for the EOS compositional model are $\ln K_i, N_i, P$, and N_w for both the fracture and matrix systems.

5.1.1 Fracture Equations

The component material balance equations for the fracture system are described in Eq. 4.6 and Eq. 4.10. The discretized forms of these equations are discussed in this section. The finite difference forms of phase equilibrium equation (Eq. 4.12) and volume constraint equation (Eq. 4.14) can be found in detail in Shi (2003).

5.1.1.1 Hydrocarbon Component Material Balance Equation

A central finite difference scheme in space and a backward finite difference scheme in time are used to discretized hydrocarbon material balance equations. Neglecting the dispersion term, the residual form of Eq. 4.6 can be written as

$$\begin{aligned}R_i^m =& V_b \frac{(\phi_i N_i)^{n+1} - (\phi_i N_i)^n}{\Delta t} \\ & - V_b \nabla \cdot \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \cdot \nabla P - V_b \nabla \cdot \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \cdot (\nabla P_{c1j} - \gamma_j \nabla D) \\ & - q_i + \tau_{mfi}\end{aligned} \qquad (5.1)$$

for $i = 1, 2, \ldots, n_c$, and $j =$oil/gas

In Eq. 5.1, the first term in the right-hand side is the accumulation term, the second term is the flux term, the third term is the capillary-gravity term, the forth term is source or sink term, and the last term is the matrix-fracture transfer term.

For a 3D case, the flux term can be expressed by

$$\nabla\cdot\xi_j x_{ij}\lambda_j\cdot\nabla P = \frac{\partial}{\partial x}\left(\xi_j x_{ij}\lambda_{jx}\frac{\partial P}{\partial x}\right)+\frac{\partial}{\partial y}\left(\xi_j x_{ij}\lambda_{jy}\frac{\partial P}{\partial y}\right)+\frac{\partial}{\partial z}\left(\xi_j x_{ij}\lambda_{jz}\frac{\partial P}{\partial z}\right) \tag{5.2}$$

Each term in Eq 5.2 needs to be discretized using the finite difference approximation in space. The discretization with respect to x yields

$$\begin{aligned}
&\frac{\partial}{\partial x}\left(\xi_j x_{ij}\lambda_{jx}\frac{\partial P}{\partial x}\right)\\
&\quad=\frac{1}{\Delta X_x}\left[\left(\xi_j x_{ij}\lambda_{jx}\frac{\partial P}{\partial x}\right)_{x+1/2}-\left(\xi_j x_{ij}\lambda_{jx}\frac{\partial P}{\partial x}\right)_{x-1/2}\right]\\
&\quad=\frac{1}{\Delta X_x}\left[(\xi_j x_{ij}\lambda_{jx})_{x+1/2}\frac{P_{x+1}-P_x}{\Delta X_{x+1/2}}\,(\xi_j x_{ij}\lambda_{jx})_{x-1/2}\frac{P_x-P_{x-1}}{\Delta X_{x-1/2}}\right]\\
&\quad=\frac{1}{\Delta X_x}\left[\left(\xi_j x_{ij}\frac{\lambda_{jx}}{\Delta X}\right)_{x+1/2}(P_{x+1}-P_x)-\left(\xi_j x_{ij}\frac{\lambda_{jx}}{\Delta X}\right)_{x-1/2}(P_x-P_{x-1})\right]
\end{aligned} \tag{5.3}$$

Using the same procedure for the discretization in the y and z directions yields

$$\begin{aligned}
&\frac{\partial}{\partial y}\left(\xi_j x_{ij}\lambda_{jx}\frac{\partial P}{\partial y}\right)\\
&\quad=\frac{1}{\Delta Y_y}\left[\left(\xi_j x_{ij}\frac{\lambda_{jy}}{\Delta Y}\right)_{y+1/2}(P_{y+1}-P_y)-\left(\xi_j x_{ij}\frac{\lambda_{jy}}{\Delta Y}\right)_{y-1/2}(P_y-P_{y-1})\right]
\end{aligned} \tag{5.4}$$

$$\begin{aligned}
&\frac{\partial}{\partial z}\left(\xi_j x_{ij}\lambda_{jz}\frac{\partial P}{\partial z}\right)\\
&\quad=\frac{1}{\Delta Z_z}\left[\left(\xi_j x_{ij}\frac{\lambda_{jz}}{\Delta Z}\right)_{z+1/2}(P_{z+1}-P_z)-\left(\xi_j x_{ij}\frac{\lambda_{jz}}{\Delta Z}\right)_{z-1/2}(P_z-P_{z-1})\right]
\end{aligned} \tag{5.5}$$

In the above equations $\lambda_j = k\left(\frac{k_{rj}}{\mu_j}\right)$ is the mobility ratio of phase j. Separating the absolute permeability from the mobility term, the discretized form of the flux term

$(\nabla \cdot \sum \xi_j x_{ij} \lambda_j \nabla P)$ for a two-phase system is expressed as

$$
\begin{aligned}
&\nabla \cdot \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \nabla P \\
&= \frac{1}{\Delta X_x} \left\{ \begin{array}{l} \left[\left(\frac{\xi_o}{\mu_o} x_i k_{ro} \right)_{x+\frac{1}{2}} + \left(\frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{x+\frac{1}{2}} \right] \left(\frac{k_x}{\Delta X} \right)_{x+\frac{1}{2}} (P_{x+1} - P_x) - \\ \left[\left(\frac{\xi_o}{\mu_o} x_i k_{ro} \right)_{x-\frac{1}{2}} + \left(\frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{x-\frac{1}{2}} \right] \left(\frac{k_x}{\Delta X} \right)_{x-\frac{1}{2}} (P_x - P_{x-1}) \end{array} \right\} \\
&+ \frac{1}{\Delta Y_y} \left\{ \begin{array}{l} \left[\left(\frac{\xi_o}{\mu_o} x_i k_{ro} \right)_{y+\frac{1}{2}} + \left(\frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{y+\frac{1}{2}} \right] \left(\frac{k_y}{\Delta Y} \right)_{y+\frac{1}{2}} (P_{y+1} - P_y) - \\ \left[\left(\frac{\xi_o}{\mu_o} x_i k_{ro} \right)_{y-\frac{1}{2}} + \left(\frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{y-\frac{1}{2}} \right] \left(\frac{k_y}{\Delta Y} \right)_{y-\frac{1}{2}} (P_y - P_{y-1}) \end{array} \right\} \\
&+ \frac{1}{\Delta Z_z} \left\{ \begin{array}{l} \left[\left(\frac{\xi_o}{\mu_o} x_i k_{ro} \right)_{z+\frac{1}{2}} + \left(\frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{z+\frac{1}{2}} \right] \left(\frac{k_z}{\Delta Z} \right)_{z+\frac{1}{2}} (P_{z+1} - P_z) - \\ \left[\left(\frac{\xi_o}{\mu_o} x_i k_{ro} \right)_{z-\frac{1}{2}} + \left(\frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{z-\frac{1}{2}} \right] \left(\frac{k_z}{\Delta Z} \right)_{z-\frac{1}{2}} (P_z - P_{z-1}) \end{array} \right\}
\end{aligned}
\tag{5.6}
$$

In the case of single-phase, either oil or gas mobility term will be included. In the simulator each gridblock is represented by one point and all properties of that gridblock is assigned to that point, any property at the interface boundaries $x \pm \frac{1}{2}, y \pm \frac{1}{2}$, and $z \pm \frac{1}{2}$ needs to be evaluated using the properties at the center of the gridblocks. For the mobility terms, the classical single-point upstream weighting is used.

$$
\begin{aligned}
\left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x-1/2} &= \left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x-1} && for \quad (\Phi_j)_{x-1} > (\Phi_j)_x \\
\left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x+1/2} &= \left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x} && for \quad (\Phi_j)_{x} > (\Phi_j)_{x+1} \\
\left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x-1/2} &= \left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x} && for \quad (\Phi_j)_{x-1} < (\Phi_j)_x \\
\left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x+1/2} &= \left(\frac{\xi_j}{\mu_j}x_{ij}k_{rj}\right)_{x+1} && for \quad (\Phi_j)_{x} < (\Phi_j)_{x+1}
\end{aligned}
\tag{5.7}
$$

where Φ_j is the potential of phase j. For the permeability terms at the interface of gridblocks, a harmonic average is used (see Appendix A for more details).

$$
\left(\frac{k}{\Delta X}\right)_{x\pm\frac{1}{2}} = \frac{2}{\frac{\Delta X_x}{k_x} + \frac{\Delta X_{x\pm1}}{k_{x\pm1}}}
\tag{5.8}
$$

Based on the potential, there are four scenarios in each direction

1. $\Phi_{jl+1} < \Phi_{jl}$ and $\Phi_{jl} < \Phi_{jl-1}$
2. $\Phi_{jl+1} > \Phi_{jl}$ and $\Phi_{jl} > \Phi_{jl-1}$
3. $\Phi_{jl+1} > \Phi_{jl}$ and $\Phi_{jl} < \Phi_{jl-1}$
4. $\Phi_{jl+1} < \Phi_{jl}$ and $\Phi_{jl} > \Phi_{jl-1}$

Assuming scenario 1 in all directions and using Eq. 5.8, Eq. 5.6 is written as

$$\begin{aligned}
&V_b \nabla \cdot \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \cdot \nabla P = \\
&\quad \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_x \tilde{k}_{x+\frac{1}{2}} \left(P_{x+1} - P_x \right) \\
&\quad - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{x-1} \tilde{k}_{x-1} \left(P_x - P_{x-1} \right) \\
&\quad + \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_y \tilde{k}_{y+\frac{1}{2}} \left(P_{y+1} - P_y \right) \\
&\quad - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{y-1} \tilde{k}_{y-\frac{1}{2}} \left(P_y - P_{y-1} \right) \\
&\quad + \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_z \tilde{k}_{z+\frac{1}{2}} \left(P_{z+1} - P_z \right) \\
&\quad - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{z-1} \tilde{k}_{z-\frac{1}{2}} \left(P_z - P_{z-1} \right)
\end{aligned} \tag{5.9}$$

where transmissibility constants ($\tilde{k}$) are defined as

$$\tilde{k}_{x\pm\frac{1}{2}} = \frac{2\Delta Y \cdot \Delta Z}{\left(\frac{\Delta X}{k}\right)_x + \left(\frac{\Delta X}{k}\right)_{x\pm1}} \tag{5.10}$$

$$\tilde{k}_{y\pm\frac{1}{2}} = \frac{2\Delta X \cdot \Delta Z}{\left(\frac{\Delta Y}{k}\right)_y + \left(\frac{\Delta Y}{k}\right)_{y\pm1}} \tag{5.11}$$

$$\tilde{k}_{z\pm\frac{1}{2}} = \frac{2\Delta X \cdot \Delta Y}{\left(\frac{\Delta Z}{k}\right)_z + \left(\frac{\Delta Z}{k}\right)_{z\pm1}} \tag{5.12}$$

The same procedure can be used to discretize the capillary pressure and gravity terms in Eq. 5.1.

$$
\begin{aligned}
&V_b \nabla \cdot \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \cdot \left(\nabla P_{c1j} - \gamma_j \nabla D\right) = \\
&\quad \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{x+\frac{1}{2}} \tilde{k}_{x+\frac{1}{2}} \left[\left(P_{c1j}^{x+1} - P_{c1j}^{x}\right) - \left(\gamma_j\right)_{x+\frac{1}{2}} \left(D_{x+1} - D_x\right)\right] \\
&\quad - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{x-\frac{1}{2}} \tilde{k}_{x-\frac{1}{2}} \left[\left(P_{c1j}^{x} - P_{c1j}^{x-1}\right) - \left(\gamma_j\right)_{x-\frac{1}{2}} \left(D_x - D_{x-1}\right)\right] \\
&\quad + \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{y+\frac{1}{2}} \tilde{k}_{y+\frac{1}{2}} \left[\left(P_{c1j}^{y+1} - P_{c1j}^{y}\right) - \left(\gamma_j\right)_{y+\frac{1}{2}} \left(D_{y+1} - D_y\right)\right] \\
&\quad - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{y-\frac{1}{2}} \tilde{k}_{y-\frac{1}{2}} \left[\left(P_{c1j}^{y} - P_{c1j}^{y-1}\right) - \left(\gamma_j\right)_{y-\frac{1}{2}} \left(D_y - D_{y-1}\right)\right] \\
&\quad + \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{z+\frac{1}{2}} \tilde{k}_{z+\frac{1}{2}} \left[\left(P_{c1j}^{z+1} - P_{c1j}^{z}\right) - \left(\gamma_j\right)_{z+\frac{1}{2}} \left(D_{z+1} - D_z\right)\right] \\
&\quad - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{z-\frac{1}{2}} \tilde{k}_{z-\frac{1}{2}} \left[\left(P_{c1j}^{z} - P_{c1j}^{z-1}\right) - \left(\gamma_j\right)_{z-\frac{1}{2}} \left(D_z - D_{z-1}\right)\right]
\end{aligned}
\tag{5.13}
$$

The only remaining term in Eq. 5.1 is the matrix-fracture transfer term defined by Eq. 4.16. The discrete form of the matrix-fracture transfer function is expressed by

$$
\tau_{mfi} = NM \cdot \sum_{l=1}^{N_b} T_{mfil} \left[\left(P_{ml} - P_f\right) + \left(P_{c1jl}^{m} - P_{c1j}^{f}\right) - \gamma_j^* \left(D_{ml} - D_f\right)\right] \tag{5.14}
$$

where NM is the number of matrix gridblocks within a fracture gridblock and N_b is the number of matrix subgrids which have surface exposed to fractures. T_{mfi} is the transmissibility depending on the geometry of the matrix subgrids. T_{mfi} include both constant part of the transmissibility and the fluid related (mobility) part of the transmissibility. For the mobility part of the transmissibility, a one-point upstream weighting is used. γ_j^* is the specific gravity of phase j and is calculated based on the one-point upstream weighting. More detail of the transfer function term is given in Section 5.3 on page 107.

Putting Eqs. 5.9, 5.13, and 5.14 together, the discretized form of the mass balance residual for component i (Eq. 5.1) is given by

$$
\begin{aligned}
R_i^m &= \frac{(V_p N_i)^{n+1} - (V_p N_i)^n}{\Delta t} \\
&- \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{x+\frac{1}{2}} \tilde{k}_{x+\frac{1}{2}} \left[(P_{x+1} - P_x) + \left(P_{c1j}^{x+1} - P_{c1j}^{x}\right) - (\gamma_j)_{x+\frac{1}{2}} (D_{x+1} - D_x)\right] \\
&+ \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{x-\frac{1}{2}} \tilde{k}_{x-\frac{1}{2}} \left[(P_x - P_{x-1}) + \left(P_{c1j}^{x} - P_{c1j}^{x-1}\right) - (\gamma_j)_{x-\frac{1}{2}} (D_x - D_{x-1})\right] \\
&- \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{y+\frac{1}{2}} \tilde{k}_{y+\frac{1}{2}} \left[(P_{y+1} - P_y) + \left(P_{c1j}^{y+1} - P_{c1j}^{y}\right) - (\gamma_j)_{y+\frac{1}{2}} (D_{y+1} - D_y)\right] \\
&+ \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{y-\frac{1}{2}} \tilde{k}_{y-\frac{1}{2}} \left[(P_y - P_{y-1}) + \left(P_{c1j}^{y} - P_{c1j}^{y-1}\right) - (\gamma_j)_{y-\frac{1}{2}} (D_y - D_{y-1})\right] \\
&- \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{z+\frac{1}{2}} \tilde{k}_{z+\frac{1}{2}} \left[(P_{z+1} - P_z) + \left(P_{c1j}^{z+1} - P_{c1j}^{z}\right) - (\gamma_j)_{z+\frac{1}{2}} (D_{z+1} - D_z)\right] \\
&+ \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg}\right)_{z-\frac{1}{2}} \tilde{k}_{z-\frac{1}{2}} \left[(P_z - P_{z-1}) + \left(P_{c1j}^{z} - P_{c1j}^{z-1}\right) - (\gamma_j)_{z-\frac{1}{2}} (D_z - D_{z-1})\right] \\
&- q_i + NM \cdot \sum_{l=1}^{N_b} T_{mfil} \left[(P_{ml} - P_f) + \left(P_{c1jl}^{m} - P_{c1j}^{f}\right) - \gamma_j^* (D_{ml} - D_f)\right]
\end{aligned}
\tag{5.15}
$$

for $i = 1, \ldots, n_c$

Note that $P_x = P_y = P_z = P_f$. In Eq. 5.15 $\tilde{k}_{x\pm\frac{1}{2}}, \tilde{k}_{y\pm\frac{1}{2}}$, and $\tilde{k}_{z\pm\frac{1}{2}}$ are constants and all other terms are functions of the primary variables $\ln K_i, N_i$, and P.

5.1.1.2 Water Component Material Balance Equation

The same procedure is used to discretize the water component residual equation (Eq. 4.10) in space and time.

$$
\begin{aligned}
R^w = & \frac{(V_pN_w)^{n+1} - (V_pN_w)^n}{\Delta t} \\
& - \left(\frac{\xi_w}{\mu_w}k_{rw}\right)_{x+\frac{1}{2}} \tilde{k}_{x+\frac{1}{2}} \left[(P_{x+1} - P_x) - (\gamma_w)_{x+\frac{1}{2}} (D_{x+1} - D_x)\right] \\
& + \left(\frac{\xi_w}{\mu_w}k_{rw}\right)_{x-\frac{1}{2}} \tilde{k}_{x-\frac{1}{2}} \left[(P_x - P_{x-1}) - (\gamma_w)_{x-\frac{1}{2}} (D_x - D_{x-1})\right] \\
& - \left(\frac{\xi_w}{\mu_w}k_{rw}\right)_{y+\frac{1}{2}} \tilde{k}_{y+\frac{1}{2}} \left[(P_{y+1} - P_y) - (\gamma_w)_{y+\frac{1}{2}} (D_{y+1} - D_y)\right] \\
& + \left(\frac{\xi_w}{\mu_w}k_{rw}\right)_{y-\frac{1}{2}} \tilde{k}_{y-\frac{1}{2}} \left[(P_y - P_{y-1}) - (\gamma_w)_{y-\frac{1}{2}} (D_y - D_{y-1})\right] \\
& - \left(\frac{\xi_w}{\mu_w}k_{rw}\right)_{z+\frac{1}{2}} \tilde{k}_{z+\frac{1}{2}} \left[(P_{z+1} - P_z) - (\gamma_w)_{z+\frac{1}{2}} (D_{z+1} - D_z)\right] \\
& + \left(\frac{\xi_w}{\mu_w}k_{rw}\right)_{z-\frac{1}{2}} \tilde{k}_{z-\frac{1}{2}} \left[(P_z - P_{z-1}) - (\gamma_w)_{z-\frac{1}{2}} (D_z - D_{z-1})\right] \\
& -q_w + NM \cdot \sum_{l=1}^{N_b} T_{mfwl} \left[(P_{ml} - P_f) - \gamma_w^* (D_{ml} - D_f)\right]
\end{aligned}
\tag{5.16}
$$

In Eq. 5.16 $\tilde{k}_{x\pm\frac{1}{2}}$, $\tilde{k}_{y\pm\frac{1}{2}}$, and $\tilde{k}_{z\pm\frac{1}{2}}$ are constants and all other terms are functions of the primary variables N_w, and P. Equations 5.15 and 5.16 are the discretized form of the material balance governing equations for the fracture system. They are highly nonlinear and need to be linearized using a Newton's method. The chain rule is used to calculate the derivatives of these equations with respect to the primary variables to form the Jacobian matrix.

5.1.1.3 Jacobian Matrix for the Fracture Media

The Jacobian is the best linear approximation to a non-linear function at a given point. To linearize the fracture governing equations, the partial derivatives of the residual equations must be calculated with respect to the primary variables. The primary variables are the fracture primary variables $\left(\mathbf{x}_f = [\ln K_{fi}, N_{fi}, P_f, N_{fw}]^T\right)$ and matrix primary variables $\left(\mathbf{x}_m = [\ln K_{mi}, N_{mi}, P_m, N_{mw}]^T\right)$. The fracture governing equations are: the phase equilibrium equation (Eq. 4.12), the volume constraint equation (Eq. 4.14), and the discrete forms of the fracture material balance equations (Eqs. 5.15, and 5.16). The volume constraint and phase equilibrium equations are only functions of the fracture primary variables $(\mathbf{x}_f)$. The Jacobian terms of these equations are the same as those in the single porosity model and can be found in detail in Shi (2003).

The partial derivatives of the material balance equations are functions of both fracture $(\mathbf{x}_f)$ and matrix $(\mathbf{x}_m)$ primary variables. The residual forms of the material balance equations can be considered as (the subscript f is dropped for the simplicity)

$$\begin{aligned} R^m &= R^m_{accum} + R^m_{conv} + R^m_{source} + R^m_{mftf} \\ R^w &= R^w_{accum} + R^w_{conv} + R^w_{source} + R^w_{mftf} \end{aligned} \tag{5.17}$$

where R^m and R^w are the hydrocarbon and water residuals of the material balance equations, respectively. The subscripts $accum, conv, source$, and $mftf$ stand for *accumulation*, *convection*, *source*, and *matrix-fracture transfer function* terms, respectively. By taking the derivatives of these equations with respect to both matrix and fracture primary variables we obtain

$$\begin{aligned} \frac{\partial R^m}{\partial \mathbf{x}} &= \frac{\partial R^m}{\partial \mathbf{x}_f} + \frac{\partial R^m}{\partial \mathbf{x}_m} = \frac{\partial R^m_{accum}}{\partial \mathbf{x}_f} + \frac{\partial R^m_{conv}}{\partial \mathbf{x}_f} + \frac{\partial R^m_{source}}{\partial \mathbf{x}_f} + \frac{\partial R^m_{mftf}}{\partial \mathbf{x}} \\ \frac{\partial R^w}{\partial \mathbf{x}} &= \frac{\partial R^w}{\partial \mathbf{x}_f} + \frac{\partial R^w}{\partial \mathbf{x}_m} = \frac{\partial R^w_{accum}}{\partial \mathbf{x}_f} + \frac{\partial R^w_{conv}}{\partial \mathbf{x}_f} + \frac{\partial R^w_{source}}{\partial \mathbf{x}_f} + \frac{\partial R^w_{mftf}}{\partial \mathbf{x}} \end{aligned} \tag{5.18}$$

where $\mathbf{x} = [\mathbf{x}_f \quad \mathbf{x}_m]^T$. As shown in Eq. 5.18, the accumulation, convection, and source terms are only functions of the fracture primary variables ($\mathbf{x}_f$). Hence, their derivatives are the same as those of the single porosity model. The matrix-fracture transfer function term is a function of both fracture and matrix primary variables. The Jacobian of material balance equations of the single porosity model need to be modified by adding $\frac{\partial R^m_{mftf}}{\partial \mathbf{x}}$ for the dual porosity model. In Section 4.3 of Chapter 4, the residual equation for the fracture system was written as

$$\bar{\bar{A}}_{ff}\mathbf{F} + \bar{\bar{B}}_{fm}\mathbf{M} = \mathbf{R}_f \tag{4.83}$$

$\bar{\bar{A}}_{ff}$, the Jacobian of fracture equations with respect to the fracture independent variables, is a block square matrix of the size $N \times N$, where N is the number of fracture gridblocks or simulation gridblocks. Each block in $\bar{\bar{A}}_{ff}$ is a $(2n_c{+}2)\times(2n_c{+}2)$ matrix. It includes derivatives of all fracture governing equations with respect to the fracture primary variables. The portion of $\bar{\bar{A}}_{ff}$ which holds the derivatives of the material balance equations can be expressed as

$$\begin{aligned}
\left[\bar{\bar{A}}_{ff}\right]_m &= \frac{\partial R^m_{accum}}{\partial \mathbf{x}_f} + \frac{\partial R^m_{conv}}{\partial \mathbf{x}_f} + \frac{\partial R^m_{source}}{\partial \mathbf{x}_f} + \frac{\partial R^m_{mftf}}{\partial \mathbf{x}_f} \\
\left[\bar{\bar{A}}_{ff}\right]_w &= \frac{\partial R^w_{accum}}{\partial \mathbf{x}_f} + \frac{\partial R^w_{conv}}{\partial \mathbf{x}_f} + \frac{\partial R^w_{source}}{\partial \mathbf{x}_f} + \frac{\partial R^w_{mftf}}{\partial \mathbf{x}_f}
\end{aligned} \tag{5.19}$$

In matrix form, $\bar{\bar{A}}_{ff}$ is expressed as

$$\bar{\bar{A}}_{ff} = \begin{bmatrix} \dfrac{\partial R_1^f}{\partial \ln K_1} & \cdots & \dfrac{\partial R_1^f}{\partial \ln K_{n_c}} & \dfrac{\partial R_1^f}{\partial N_1} & \cdots & \dfrac{\partial R_1^f}{\partial N_{n_c}} & \dfrac{\partial R_1^f}{\partial P} & \dfrac{\partial R_1^f}{\partial N_w} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial R_{n_c}^f}{\partial \ln K_1} & \cdots & \dfrac{\partial R_{n_c}^f}{\partial \ln K_{n_c}} & \dfrac{\partial R_{n_c}^f}{\partial N_1} & \cdots & \dfrac{\partial R_{n_c}^f}{\partial N_{n_c}} & \dfrac{\partial R_{n_c}^f}{\partial P} & \dfrac{\partial R_{n_c}^f}{\partial N_w} \\ \\ \dfrac{\partial R^v}{\partial \ln K_1} & \cdots & \dfrac{\partial R^v}{\partial \ln K_{n_c}} & \dfrac{\partial R^v}{\partial N_1} & \cdots & \dfrac{\partial R^v}{\partial N_{n_c}} & \dfrac{\partial R^v}{\partial P} & \dfrac{\partial R^v}{\partial N_w} \\ \\ \dfrac{\partial R_1^m}{\partial \ln K_1} & \cdots & \dfrac{\partial R_1^m}{\partial \ln K_{n_c}} & \dfrac{\partial R_1^m}{\partial N_1} & \cdots & \dfrac{\partial R_1^m}{\partial N_{n_c}} & \dfrac{\partial R_1^m}{\partial P} & \dfrac{\partial R_1^m}{\partial N_w} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial R_{n_c}^m}{\partial \ln K_1} & \cdots & \dfrac{\partial R_{n_c}^m}{\partial \ln K_{n_c}} & \dfrac{\partial R_{n_c}^m}{\partial N_1} & \cdots & \dfrac{\partial R_{n_c}^m}{\partial N_{n_c}} & \dfrac{\partial R_{n_c}^m}{\partial P} & \dfrac{\partial R_{n_c}^m}{\partial N_w} \\ \\ \dfrac{\partial R^w}{\partial \ln K_1} & \cdots & \dfrac{\partial R^w}{\partial \ln K_{n_c}} & \dfrac{\partial R^w}{\partial N_1} & \cdots & \dfrac{\partial R^w}{\partial N_{n_c}} & \dfrac{\partial R^w}{\partial P} & \dfrac{\partial R^w}{\partial N_w} \end{bmatrix} \tag{5.20}$$

Matrix $\bar{\bar{B}}_{fm}$ in Eq. 4.83 , Jacobian of fracture equations with respect to the matrix independent variables, is a matrix of the size $(2n_c+2) \times N_b(2n_c+2)$ where N_b is the number of matrix subgrids within a fracture gridblock. It is the matrix of partial derivatives of matrix-fracture transfer function with respect to the matrix primary variables.

$$\begin{aligned} \left[\bar{\bar{B}}_{fm}\right]_m &= \frac{\partial R_{mftf}^m}{\partial \mathbf{x}_m} \\ \left[\bar{\bar{B}}_{fm}\right]_w &= \frac{\partial R_{mftf}^w}{\partial \mathbf{x}_m} \end{aligned} \tag{5.21}$$

In matrix form, $\bar{\bar{B}}_{fm}$ is expressed as

$$\bar{\bar{B}}_{fm} = \begin{bmatrix} \dfrac{\partial R_1^f}{\partial \mathbf{x}_1} & \dfrac{\partial R_1^f}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R_1^f}{\partial \mathbf{x}_{N_b}} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial R_{n_c}^f}{\partial \mathbf{x}_1} & \dfrac{\partial R_{n_c}^f}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R_{n_c}^f}{\partial \mathbf{x}_{N_b}} \\ \\ \dfrac{\partial R^v}{\partial \mathbf{x}_1} & \dfrac{\partial R^v}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R^v}{\partial \mathbf{x}_{N_b}} \\ \\ \dfrac{\partial R_1^m}{\partial \mathbf{x}_1} & \dfrac{\partial R_1^m}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R_1^m}{\partial \mathbf{x}_{N_b}} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial R_{n_c}^m}{\partial \mathbf{x}_1} & \dfrac{\partial R_{n_c}^m}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R_{n_c}^m}{\partial \mathbf{x}_{N_b}} \\ \\ \dfrac{\partial R^w}{\partial \mathbf{x}_1} & \dfrac{\partial R^w}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R^w}{\partial \mathbf{x}_{N_b}} \end{bmatrix} \tag{5.22}$$

where $\mathbf{x}_j = [\ln K_i, N_i, P, N_w]_j^T$ is the vector of the primary variables of the matrix subgrid j and N_b is the total number of the matrix subgrids. Only the matrix subgrids which have a surface exposed to the fracture will have a corresponding Jacobian in $\bar{\bar{B}}_{fm}$.

5.1.2 Matrix Equations

A method similar to the one used to discretize the fracture equations is used here for discretizing the matrix equations. The component material balance equations for the matrix system are described in Eq. 4.7 and Eq. 4.11.

5.1.2.1 Hydrocarbon Component Material Balance Equation

The hydrocarbon residual equation for the matrix system is given as

$$R_i^m = V_b \frac{N_i^{n+1} - N_i^n}{\Delta t} - V_b \nabla \cdot \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \cdot \nabla P - V_b \nabla \cdot \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \cdot (\nabla P_{c1j} - \gamma_j \nabla D) \tag{5.23}$$

The MINC method (Pruess and Narasimhan, 1985) is used for discretization in the space. The advantage of the MINC method is that it reduces the dimension of the problem by one dimension in the space.

$$\nabla \cdot \xi_j x_{ij} \lambda_j \cdot \nabla P = \frac{\partial}{\partial h}\left(\xi_j x_{ij} \lambda_{jh} \frac{\partial P}{\partial h}\right) + \frac{\partial}{\partial z}\left(\xi_j x_{ij} \lambda_{jz} \frac{\partial P}{\partial z}\right) \tag{5.24}$$

The derivative in the horizontal direction is approximated by (see Appendix B for more details)

$$\frac{\partial}{\partial h}\left(\xi_j x_{ij} \lambda_{jh} \frac{\partial P}{\partial h}\right) = \frac{2\Delta Z_{hk}}{V_{bhk}} \left[\begin{array}{l} (\xi_j x_{ij} \lambda_{jx})_{h+\frac{1}{2}k} Y_{h+\frac{1}{2}k} \dfrac{P_{h+1k} - P_{hk}}{X_{h+1k} - X_{hk}} - (\xi_j x_{ij} \lambda_{jx})_{h-\frac{1}{2}k} Y_{h-\frac{1}{2}k} \dfrac{P_{hk} - P_{h-1k}}{X_{hk} - X_{h-1k}} \\ + (\xi_j x_{ij} \lambda_{jy})_{h+\frac{1}{2}k} X_{h+\frac{1}{2}k} \dfrac{P_{h+1k} - P_{hk}}{Y_{h+1k} - Y_{hk}} - (\xi_j x_{ij} \lambda_{jy})_{h-\frac{1}{2}k} X_{h-\frac{1}{2}k} \dfrac{P_{hk} - P_{h-1k}}{Y_{hk} - Y_{h-1k}} \end{array} \right] \tag{5.25}$$

And the derivative in the vertical direction is approximated by

$$\frac{\partial}{\partial z}\left(\xi_j x_{ij} \lambda_{jz} \frac{\partial P}{\partial z}\right) = \frac{2}{\Delta Z_{hk}} \left[(\xi_j x_{ij} \lambda_{jz})_{hk+\frac{1}{2}} \frac{P_{hk+1} - P_{hk}}{\Delta Z_{hk+1} + \Delta Z_{hk}} - (\xi_j x_{ij} \lambda_{zx})_{hk-\frac{1}{2}} \frac{P_{hk} - P_{hk-1}}{\Delta Z_{hk} + \Delta Z_{hk-1}} \right] \tag{5.26}$$

Separating the absolute permeability from the mobility term, the discrete form of the flux term is given as

$$
\begin{aligned}
&V_b \nabla \cdot \xi_j x_{ij} \lambda_j \cdot \nabla P = \\
&2\Delta Z_{hk} \left\{ \begin{array}{l} \quad \left[\left(\frac{\xi_j}{\mu_j} x_{ij} k_{rj} \right)_{h+\frac{1}{2}k} \left(\frac{k_{h+\frac{1}{2}k} Y_{h+\frac{1}{2}k}}{X_{h+1k} - X_{hk}} + \frac{k_{h+\frac{1}{2}k} X_{h+\frac{1}{2}k}}{Y_{h+1k} - Y_{hk}} \right) \right] (P_{h+1k} - P_{hk}) \\ - \left[\left(\frac{\xi_j}{\mu_j} x_{ij} k_{rj} \right)_{h-\frac{1}{2}k} \left(\frac{k_{h-\frac{1}{2}k} Y_{h-\frac{1}{2}k}}{X_{hk} - X_{h-1k}} + \frac{k_{h-\frac{1}{2}k} X_{h-\frac{1}{2}k}}{Y_{hk} - Y_{h-1k}} \right) \right] (P_{hk} - P_{h-1k}) \end{array} \right\} \\
&+ 2\Delta X_{hk} \Delta Y_{hk} \left[\begin{array}{l} \quad \left(\frac{\xi_j}{\mu_j} x_{ij} k_{rj} \right)_{hk+\frac{1}{2}} \left(\frac{k_{hk+\frac{1}{2}}}{\Delta Z_{hk+1} + \Delta Z_{hk}} \right) (P_{hk+1} - P_{hk}) \\ - \left(\frac{\xi_j}{\mu_j} x_{ij} k_{rj} \right)_{hk-\frac{1}{2}} \left(\frac{k_{hk-\frac{1}{2}}}{\Delta Z_{hk} + \Delta Z_{hk-1}} \right) (P_{hk} - P_{hk-1}) \end{array} \right]
\end{aligned} \tag{5.27}
$$

The constant parts of the transmissibilities in the horizontal and vertical direction are defined as

$$
\tilde{k}_{h+\frac{1}{2}k} = 2\Delta Z_{hk} \left(\frac{k_{h+\frac{1}{2}k} Y_{h+\frac{1}{2}k}}{X_{h+1k} - X_{hk}} + \frac{k_{h+\frac{1}{2}k} X_{h+\frac{1}{2}k}}{Y_{h+1k} - Y_{hk}} \right) \tag{5.28}
$$

$$
\tilde{k}_{hk+\frac{1}{2}} = 2\Delta X_{hk} \Delta Y_{hk} \left(\frac{k_{hk+\frac{1}{2}}}{\Delta Z_{hk+1} + \Delta Z_{hk}} \right) \tag{5.29}
$$

Equation 5.23, following substitution of Eq. 5.27 and considering capillary pressure and gravity terms, can be written as

$$
\begin{aligned}
R_i^m = {} & \frac{(V_p N_i)^{n+1} - (V_p N_i)^n}{\Delta t} \\
& - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{h+\frac{1}{2}k} \tilde{k}_{h+\frac{1}{2}k} \left[\begin{array}{l} (P_{h+1k} - P_{hk}) + \left(P_{c1j}^{h+1k} - P_{c1j}^{hk} \right) \\ - (\gamma_j)_{h+\frac{1}{2}k} (D_{h+1k} - D_{hk}) \end{array} \right] \\
& + \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{h-\frac{1}{2}k} \tilde{k}_{h-\frac{1}{2}k} \left[\begin{array}{l} (P_{hk} - P_{h-1k}) + \left(P_{c1j}^{hk} - P_{c1j}^{h-1k} \right) \\ - (\gamma_j)_{h-\frac{1}{2}k} (D_{hk} - D_{h-1k}) \end{array} \right] \\
& - \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{hk+\frac{1}{2}} \tilde{k}_{hk+\frac{1}{2}} \left[\begin{array}{l} (P_{kk+1} - P_{hk}) + \left(P_{c1j}^{hk+1} - P_{c1j}^{hk} \right) \\ - (\gamma_j)_{hk+\frac{1}{2}} (D_{hk+1} - D_{hk}) \end{array} \right] \\
& + \left(\frac{\xi_o}{\mu_o} x_i k_{ro} + \frac{\xi_g}{\mu_g} y_i k_{rg} \right)_{hk-\frac{1}{2}} \tilde{k}_{hk-\frac{1}{2}} \left[\begin{array}{l} (P_{hk} - P_{hk-1}) + \left(P_{c1j}^{hk} - P_{c1j}^{hk-1} \right) \\ - (\gamma_j)_{hk-\frac{1}{2}} (D_{hk} - D_{hk-1}) \end{array} \right]
\end{aligned}
\tag{5.30}
$$

for $i = 1, \ldots, n_c$.

In Eq. 5.30, $\tilde{k}_{h\pm\frac{1}{2}k}$ and $\tilde{k}_{hk\pm\frac{1}{2}}$ are constants and all other terms are functions of the primary variables $\ln K_i, N_i$, and P.

5.1.2.2 Water Component Material Balance Equation

A similar method to the one used to discretize the hydrocarbon material balance equation of the matrix system is used here for discretizing the water material balance equation.

$$
\begin{aligned}
R^w = {} & \frac{(V_p N_w)^{n+1} - (V_p N_w)^n}{\Delta t} \\
& - \left(\frac{\xi_w}{\mu_w} k_{rw}\right)_{h+\frac{1}{2}k} \tilde{k}_{h+\frac{1}{2}k} \left[(P_{h+1k} - P_{hk}) - (\gamma_w)_{h+\frac{1}{2}k} (D_{h+1k} - D_{hk})\right] \\
& + \left(\frac{\xi_w}{\mu_w} k_{rw}\right)_{h-\frac{1}{2}k} \tilde{k}_{h-\frac{1}{2}k} \left[(P_{hk} - P_{h-1k}) - (\gamma_w)_{h-\frac{1}{2}k} (D_{hk} - D_{h-1k})\right] \\
& - \left(\frac{\xi_w}{\mu_w} k_{rw}\right)_{hk+\frac{1}{2}} \tilde{k}_{hk+\frac{1}{2}} \left[(P_{kk+1} - P_{hk}) - (\gamma_w)_{hk+\frac{1}{2}} (D_{hk+1} - D_{hk})\right] \\
& + \left(\frac{\xi_w}{\mu_w} k_{rw}\right)_{hk-\frac{1}{2}} \tilde{k}_{hk-\frac{1}{2}} \left[(P_{hk} - P_{hk-1}) - (\gamma_w)_{hk-\frac{1}{2}} (D_{hk} - D_{hk-1})\right]
\end{aligned}
\tag{5.31}
$$

In Eq. 5.31, $\tilde{k}_{h\pm\frac{1}{2}k}$ and $\tilde{k}_{hk\pm\frac{1}{2}}$ are constants and all other terms are functions of the primary variables N_w, and P.

If the subdomain is at the top of the matrix block then P_{hk-1} is replaced by the the fracture pressure (P_f) and if the subdomain is at the bottom of the matrix block then P_{hk+1} is replaced by the fracture pressure (P_f). If the subdomain is the outermost ring then P_{h+1k} is replaced by P_f. Also, in these cases the appropriate transmisiibilities are used.

Note that, calculation of the relative permeability at the interface is based on the phase potential continuity at the boundary. The relative permeability at the interface is determined from the matrix relative permeability curve at the saturation that satisfies the continuity of capillary pressure (Chen, 1993). The wetting-phase relative permeability is give as

$$k_{rw} = \left[\omega k_{rwm} + S_{wf}(1-\omega) k_{rwm}|_{S_{wj}}\right] \tag{5.32}$$

and the non-wetting-phase relative permeability is given as

$$k_{rnw} = [\omega k_{rnw} + (1-\omega) S_{nwf} k_{rnwm}] \tag{5.33}$$

where ω is the one point upstream weighting parameter. $\omega = 1$ when the flow is from the matrix to the fracture and $\omega = 0$ when the flow is from the fracture to the matrix. When the fracture is the upstream and the flow is from the fracture to the matrix, the relative permeability is multiplied by the phase saturation in the fracture to account for the partial coverage (Chen, 1993).

5.1.2.3 Jacobian Matrix for the Matrix Media

The matrix governing equations are: the phase equilibrium equation (Eq. 4.13), the volume constraint equation (Eq. 4.15) , and the discrete forms of the matrix material balance equations (Eqs. 5.30, and 5.31). The volume constraint and phase equilibrium equations are only functions of the matrix primary variables ($\mathbf{x}_m$). The Jacobian terms of these equations are the same as those for the fracture system.

The partial derivatives of the material balance equations are functions of both fracture ($\mathbf{x}_f$) and matrix ($\mathbf{x}_m$) primary variables. The residual forms of the material balance equations can be considered as (the subscript m is dropped for the simplicity)

$$\begin{aligned} R^m &= R^m_{accum} + R^m_{conv} \\ R^w &= R^w_{accum} + R^w_{conv} \end{aligned} \tag{5.34}$$

where R^m and R^w are the hydrocarbon and water residuals of the material balance equations, respectively. By taking the derivatives of these equations with respect to both matrix and fracture primary variables we obtain

$$\begin{aligned} \frac{\partial R^m}{\partial \mathbf{x}} &= \frac{\partial R^m}{\partial \mathbf{x}_f} + \frac{\partial R^m}{\partial \mathbf{x}_m} = \frac{\partial R^m_{accum}}{\partial \mathbf{x}_m} + \frac{\partial R^m_{conv}}{\partial \mathbf{x}} \\ \frac{\partial R^w}{\partial \mathbf{x}} &= \frac{\partial R^w}{\partial \mathbf{x}_f} + \frac{\partial R^w}{\partial \mathbf{x}_m} = \frac{\partial R^w_{accum}}{\partial \mathbf{x}_m} + \frac{\partial R^w_{conv}}{\partial \mathbf{x}} \end{aligned} \tag{5.35}$$

The accumulation term is only a function of the matrix primary variables while the convection term is a function of both matrix and fracture primary variables. The convection term depends on the fracture primary variable for the subdomains that are exposed to the fracture.

Recall from Section 4.3 in Chapter 4, the residual equation for the matrix system was written as

$$\bar{\bar{C}}_{mf}\mathbf{F} + \bar{\bar{D}}_{mm}\mathbf{M} = \mathbf{R}_m \quad \text{Matrix equations} \tag{4.84}$$

$\bar{\bar{D}}_{mm}$, Jacobian of the matrix equations with respect to the matrix independent variables, is a block square matrix of the size $N_b \times N_b$ where N_b is the number of matrix subdomains within a fracture gridblock. Each block in $\bar{\bar{D}}_{mm}$ is a $(2n_c + 2) \times (2n_c + 2)$ matrix. It includes derivatives of all matrix governing equations with respect to the matrix primary variables. The portion of $\bar{\bar{D}}_{mm}$ which holds the derivatives of the material balance equations can be expressed as

$$\begin{aligned} \left[\bar{\bar{D}}_{mm}\right]_m &= \frac{\partial R^m_{accum}}{\partial \mathbf{x}_m} + \frac{\partial R^m_{conv}}{\partial \mathbf{x}_m} \\ \left[\bar{\bar{D}}_{mm}\right]_w &= \frac{\partial R^w_{accum}}{\partial \mathbf{x}_m} + \frac{\partial R^w_{conv}}{\partial \mathbf{x}_m} \end{aligned} \tag{5.36}$$

$\bar{\bar{D}}_{mm}$ has the same structure of $\bar{\bar{A}}_{ff}$ (Eq. 5.20). In Eq. 4.84 $\bar{\bar{C}}_{mf}$, Jacobian of the matrix equations with respect to the fracture independent variables, is a matrix of the size $N_b(2n_c + 2) \times (2n_c + 2)$.

$$\begin{aligned} \left[\bar{\bar{C}}_{mf}\right]_m &= \frac{\partial R^m_{conv}}{\partial \mathbf{x}_f} \\ \left[\bar{\bar{C}}_{mf}\right]_w &= \frac{\partial R^w_{conv}}{\partial \mathbf{x}_f} \end{aligned} \tag{5.37}$$

Note that the only non-zero entries in $\bar{\bar{C}}_{mf}$ come from the derivatives of residual equations of subdomains that have a face exposed to the fracture.

In the matrix form, $\bar{\bar{C}}_{mf}$ is expressed as

$$\bar{\bar{C}}_{mf} = \begin{bmatrix} \dfrac{\partial \mathbf{R}_1}{\partial \mathbf{x}_f} \\ \dfrac{\partial \mathbf{R}_2}{\partial \mathbf{x}_f} \\ \vdots \\ \dfrac{\partial \mathbf{R}_{N_b}}{\partial \mathbf{x}_f} \end{bmatrix} \tag{5.38}$$

where $\mathbf{R}_i = \left[R^f_{i1}, \ldots, R^f_{in_c}, R^v_i, R^m_{i1}, \ldots, R^m_{in_c}, R^w_i\right]^T$.

5.2 Numerical Formulation for Chemical Model

This section covers the discretization and linearization of Eq. 4.49 through Eq. 4.52. The primary variables for the chemical compositional model are N_i, P, and N_w for both fracture and matrix systems. There is no phase equilibrium governing equation in the chemical model because the gas phase is not considered in the current implementation of the chemical model in GPAS.

5.2.1 Fracture Equations

The discretized form of the component material balance equations for chemical dual porosity model are covered in this section. The residual form of the volume constraint depends only on the primary variables at its own gridblock. The single porosity form of the volume constraint is applicable for the dual porosity model. Hence, this section only covers the discretization of the material balance equations.

5.2.1.1 Hydrocarbon Component Material Balance Equation

A central finite differencing in space and a backward finite differencing in time are used to discretized chemical material balance equations. Equations 5.3, 5.4, and 5.5 are used for space discretization. The discrete form of the residual of material balance equation for a hydrocarbon component i (Eq. 4.49) at a gridblock (I, J, K) can be defined as

$$\begin{aligned}
R_i^m = & \left[\frac{V_b}{\Delta t}\left((\phi N_i)^{n+1} - (\phi N_i)^n\right)\right]_{(I,J,K)} \\
& - \sum_l \left(\frac{\xi_2}{\mu_2} x_{i2} k_{r2}\right)_{l+\frac{1}{2}} \tilde{k}_{l+\frac{1}{2}} \left[(P_{jl+1} - P_{jl}) + \gamma_{2l+\frac{1}{2}}\left(D_{l+1} - D_l\right)\right] \\
& + \sum_l \left(\frac{\xi_2}{\mu_2} x_{i2} k_{r2}\right)_{l-\frac{1}{2}} \tilde{k}_{l-\frac{1}{2}} \left[(P_{jl} - P_{jl-1}) + \gamma_{2l-\frac{1}{2}}\left(D_l - D_{l-1}\right)\right] \\
& - \sum_l \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{l+\frac{1}{2}} \tilde{k}_{l+\frac{1}{2}} \left[(P_{jl+1} - P_{jl}) + \gamma_{3l+\frac{1}{2}}\left(D_{l+1} - D_l\right)\right] \\
& + \sum_l \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{l-\frac{1}{2}} \tilde{k}_{l-\frac{1}{2}} \left[(P_{jl} - P_{jl-1}) + \gamma_{3l-\frac{1}{2}}\left(D_l - D_{l-1}\right)\right] \\
& - q_i + NM \cdot \sum_{s=1}^{N_b} T_{mfis}\left[(P_{jms} - P_{jf}) - \gamma_j^*\left(D_{ms} - D_f\right)\right] \\
& \text{for } l = I, J, K \\
& i = 1, \ldots, n_c
\end{aligned} \tag{5.39}$$

where $P_j = P + P_{cj}$ and

$$\phi = \phi_r\left[1 + c_f\left(P - P^0\right)\right] \tag{5.40}$$

The parameter c_f is the rock compressibility factor measured at a reference pressure P^0. All properties at the interface boundaries $I \pm \frac{1}{2}, J \pm \frac{1}{2}$, and $K \pm \frac{1}{2}$ need to be evaluated using the properties at the center of the gridblock. For the mobility terms, the classical single-point upstream weighting is used. A harmonic average method is used to calculate the permeability at the interface.

$$\tilde{k}_{I\pm\frac{1}{2}} = \frac{2\Delta y \Delta z k_x k_{x_{I\pm1}}}{\Delta x_{I\pm1} k_x + \Delta x k_{x_{I\pm1}}} \tag{5.41}$$

$$\tilde{k}_{J\pm\frac{1}{2}} = \frac{2\Delta x \Delta z k_y k_{y_{J\pm1}}}{\Delta y_{J\pm1} k_y + \Delta y k_{y_{J\pm1}}} \tag{5.42}$$

$$\tilde{k}_{K\pm\frac{1}{2}} = \frac{2\Delta x \Delta y k_z k_{z_{K\pm1}}}{\Delta z_{K\pm1} k_z + \Delta z k_{z_{K\pm1}}} \tag{5.43}$$

In Eq. 5.39 $\tilde{k}_{I\pm\frac{1}{2}}, \tilde{k}_{J\pm\frac{1}{2}}$, and $\tilde{k}_{K\pm\frac{1}{2}}$ are constants and all other terms are functions of the primary variables N_i, and P.

5.2.1.2 Aqueous Component Material Balance Equation

The discrete form of the residual of material balance equation for an aqueous component i (Eq. 4.51) at a gridblock (I, J, K) can be defined as

$$\begin{aligned} R_i^m = & \left[\frac{V_b}{\Delta t}\left((\phi N_i)^{n+1} - (\phi N_i)^n\right)\right]_{(I,J,K)} \\ & - \sum_l \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{l+\frac{1}{2}} \tilde{k}_{l+\frac{1}{2}} \left[(P_{jl+1} - P_{jl}) + \gamma_{3l+\frac{1}{2}}(D_{l+1} - D_l)\right] \\ & + \sum_l \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{l-\frac{1}{2}} \tilde{k}_{l-\frac{1}{2}} \left[(P_{jl} - P_{jl-1}) + \gamma_{3l-\frac{1}{2}}(D_l - D_{l-1})\right] \\ & - q_i + NM \cdot \sum_{s=1}^{N_b} T_{mfis}\left[(P_{jms} - P_{jf}) - \gamma_j^*(D_{ms} - D_f)\right] \\ & \text{for } l = I, J, K \\ & i = n_c + 1, \ldots, n_c + n_a + 1 \end{aligned} \tag{5.44}$$

In Eq. 5.44 $\tilde{k}_{I\pm\frac{1}{2}}, \tilde{k}_{J\pm\frac{1}{2}}$, and $\tilde{k}_{K\pm\frac{1}{2}}$ are constants and all other terms are functions of the primary variables N_i, and P. Equations 5.39 and 5.44 are the discretized form of the material balance governing equations for the fracture system. They are highly nonlinear and need to be linearized using a Newton's method. The chain rule is used to calculate the derivatives of these equations with respect to the primary variables to form the Jacobian matrix.

5.2.1.3 Jacobian Matrix for the Fracture Media

To linearize the fracture governing equations, the partial derivatives of the residual equations must be calculated with respect to the primary variables. The primary variables are the fracture primary variables $\left(\mathbf{x}_f = [N_{fi}, P_f, N_{fw}]^T\right)$ and matrix primary variables $\left(\mathbf{x}_m = [N_{mi}, P_m, N_{mw}]^T\right)$. The governing equations of the fracture system in the chemical model are the volume constraint equation (Eq. 4.53) and the discretized forms of the material balance equations (Eqs. 5.39 and 5.44). The volume constraint equation is only a function of the fracture primary variables ($\mathbf{x}_f$). Hence its Jacobian is the same as the Jacobian of the volume constraint equation in the single porosity model.

The partial derivatives of the material balance equations are functions of both fracture ($\mathbf{x}_f$) and matrix ($\mathbf{x}_m$) primary variables. The procedure developed in Section 5.1.1.3 is used here to construct the Jacobian matrices ($\bar{\bar{A}}_{ff}, \bar{\bar{B}}_{fm}, \bar{\bar{C}}_{mf}$, and $\bar{\bar{D}}_{mm}$).

$\bar{\bar{A}}_{ff}$, the Jacobian of the fracture equations with respect to the fracture independent variables, is a block square matrix of the size $N \times N$, where N is the number of fracture gridblocks or simulation gridblocks. Each block in $\bar{\bar{A}}_{ff}$ is a $(n_c + n_a + 2) \times (n_c + n_a + 2)$ matrix. It includes derivatives of all fracture governing equations with respect to the fracture primary variables. In matrix form $\bar{\bar{A}}_{ff}$ is expressed as

$$\bar{\bar{A}}_{ff} = \begin{bmatrix} \dfrac{\partial R^v}{\partial N_1} & \cdots & \dfrac{\partial R^v}{\partial N_{n_c+n_a}} & \dfrac{\partial R^v}{\partial P} & \dfrac{\partial R^v}{\partial N_w} \\ \\ \dfrac{\partial R^m_1}{\partial N_1} & \cdots & \dfrac{\partial R^m_1}{\partial N_{n_c+n_a}} & \dfrac{\partial R^m_1}{\partial P} & \dfrac{\partial R^m_1}{\partial N_w} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial R^m_{n_c+n_a}}{\partial N_1} & \cdots & \dfrac{\partial R^m_{n_c+n_a}}{\partial N_{n_c+n_a}} & \dfrac{\partial R^m_{n_c}}{\partial P} & \dfrac{\partial R^m_{n_c}}{\partial N_w} \\ \\ \dfrac{\partial R^w}{\partial N_1} & \cdots & \dfrac{\partial R^w}{\partial N_{n_c+n_a}} & \dfrac{\partial R^w}{\partial P} & \dfrac{\partial R^w}{\partial N_w} \end{bmatrix} \tag{5.45}$$

Matrix $\bar{\bar{B}}_{fm}$ in Eq. 4.83, Jacobian of the fracture equations with respect to the matrix independent variables, is a matrix of the size $(n_c+n_a+2)\times N_b(n_c+n_a+2)$ where N_b is the number of matrix subgrids within a fracture gridblock. It is the matrix of partial derivatives of matrix-fracture transfer function with respect to the matrix primary variables.

$$\begin{aligned} \left[\bar{\bar{B}}_{fm}\right]_m &= \frac{\partial R^m_{mftf}}{\partial \mathbf{x}_m} \\ \left[\bar{\bar{B}}_{fm}\right]_w &= \frac{\partial R^w_{mftf}}{\partial \mathbf{x}_m} \end{aligned} \tag{5.21}$$

In matrix form, $\bar{\bar{B}}_{fm}$ is expressed as

$$\bar{\bar{B}}_{fm} = \begin{bmatrix} \dfrac{\partial R^v}{\partial \mathbf{x}_1} & \dfrac{\partial R^v}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R^v}{\partial \mathbf{x}_{N_b}} \\ \\ \dfrac{\partial R^m_1}{\partial \mathbf{x}_1} & \dfrac{\partial R^m_1}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R^m_1}{\partial \mathbf{x}_{N_b}} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial R^m_{n_c+n_a}}{\partial \mathbf{x}_1} & \dfrac{\partial R^m_{n_c+n_a}}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R^m_{n_c+n_a}}{\partial \mathbf{x}_{N_b}} \\ \\ \dfrac{\partial R^w}{\partial \mathbf{x}_1} & \dfrac{\partial R^w}{\partial \mathbf{x}_2} & \cdots & \dfrac{\partial R^w}{\partial \mathbf{x}_{N_b}} \end{bmatrix} \tag{5.46}$$

where $\mathbf{x}_j = [N_i, P, N_w]_j^T$ is the vector of the primary variables of the matrix subgrid j and N_b is the total number of the matrix subgrids. Only the matrix subgrids which have a surface exposed to the fracture will have a corresponding Jacobian in $\bar{\bar{B}}_{fm}$.

5.2.2 Matrix Equations

A method similar to the one used to discretize the matrix equations of EOS compositional model is used here for discretizing the matrix equations. The component material balance equations for the fracture matrix are described in Eq. 4.50 and Eq. 4.52.

5.2.2.1 Hydrocarbon Component Material Balance Equation

The MINC method is used for discretization in the space. The discrete form of the residual of material balance equation for hydrocarbon component i at a subgrid of

(M, N) is defined as

$$\begin{aligned}
R_i^m = & \left[\frac{V_b}{\Delta t}\left((\phi N_i)^{n+1} - (\phi N_i)^n\right)\right]_{(M,N)} \\
& - \left(\frac{\xi_2}{\mu_2} x_{i2} k_{r2}\right)_{h+\frac{1}{2}} \tilde{k}_{h+\frac{1}{2}} \left[(P_{jh+1} - P_{jh}) + \gamma_{2h+\frac{1}{2}}(D_{h+1} - D_h)\right] \\
& + \left(\frac{\xi_2}{\mu_2} x_{i2} k_{r2}\right)_{h-\frac{1}{2}} \tilde{k}_{h-\frac{1}{2}} \left[(P_{jh} - P_{jh-1}) + \gamma_{2h-\frac{1}{2}}(D_h - D_{h-1})\right] \\
& - \left(\frac{\xi_2}{\mu_2} x_{i2} k_{r2}\right)_{k+\frac{1}{2}} \tilde{k}_{k+\frac{1}{2}} \left[(P_{jk+1} - P_{jk}) + \gamma_{2k+\frac{1}{2}}(D_{k+1} - D_k)\right] \\
& + \left(\frac{\xi_2}{\mu_2} x_{i2} k_{r2}\right)_{k-\frac{1}{2}} \tilde{k}_{k-\frac{1}{2}} \left[(P_{jk} - P_{jk-1}) + \gamma_{2k-\frac{1}{2}}(D_k - D_{k-1})\right] \\
& - \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{h+\frac{1}{2}} \tilde{k}_{h+\frac{1}{2}} \left[(P_{jh+1} - P_{jh}) + \gamma_{3h+\frac{1}{2}}(D_{h+1} - D_h)\right] \qquad (5.47) \\
& + \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{h-\frac{1}{2}} \tilde{k}_{h-\frac{1}{2}} \left[(P_{jh} - P_{jh-1}) + \gamma_{3h-\frac{1}{2}}(D_h - D_{h-1})\right] \\
& - \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{k+\frac{1}{2}} \tilde{k}_{k+\frac{1}{2}} \left[(P_{jk+1} - P_{jk}) + \gamma_{3k+\frac{1}{2}}(D_{k+1} - D_k)\right] \\
& + \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{k-\frac{1}{2}} \tilde{k}_{k-\frac{1}{2}} \left[(P_{jk} - P_{jk-1}) + \gamma_{3k-\frac{1}{2}}(D_k - D_{k-1})\right] \\
& - q_i + NM \cdot \sum_{s=1}^{N_b} T_{mfis} \left[(P_{jms} - P_{jf}) - \gamma_j^* (D_{ms} - D_f)\right]
\end{aligned}$$

for $i = 1, \ldots, n_c$.

In Eq. 5.47, $\tilde{k}_{h\pm\frac{1}{2}}$ and $\tilde{k}_{k\pm\frac{1}{2}}$ are constants and all other terms are functions of the primary variables N_i and P.

5.2.2.2 Aqueous Component Material Balance Equation

A similar method to the one used to discretize the hydrocarbon material balance equation is used here for discretizing the water material balance equation.

$$
\begin{aligned}
R_i^m = & \left[\frac{V_b}{\Delta t}\left((\phi N_i)^{n+1} - (\phi N_i)^n\right)\right]_{(M,N)} \\
& - \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{h+\frac{1}{2}} \tilde{k}_{h+\frac{1}{2}} \left[(P_{jh+1} - P_{jh}) + \gamma_{3h+\frac{1}{2}} (D_{h+1} - D_h)\right] \\
& + \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{h-\frac{1}{2}} \tilde{k}_{h-\frac{1}{2}} \left[(P_{jh} - P_{jh-1}) + \gamma_{3h-\frac{1}{2}} (D_h - D_{h-1})\right] \\
& - \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{k+\frac{1}{2}} \tilde{k}_{k+\frac{1}{2}} \left[(P_{jk+1} - P_{jk}) + \gamma_{3k+\frac{1}{2}} (D_{k+1} - D_k)\right] \\
& + \left(\frac{\xi_3}{\mu_3} x_{i3} k_{r3}\right)_{k-\frac{1}{2}} \tilde{k}_{k-\frac{1}{2}} \left[(P_{jk} - P_{jk-1}) + \gamma_{3k-\frac{1}{2}} (D_k - D_{k-1})\right] \\
& - q_i + NM \cdot \sum_{s=1}^{N_b} T_{mfis} \left[(P_{jms} - P_{jf}) - \gamma_j^* (D_{ms} - D_f)\right] \\
& \text{for } l = I, J, K \\
& i = n_c + 1, \ldots, n_c + n_a + 1
\end{aligned}
\tag{5.48}
$$

In Eq. 5.48 $\tilde{k}_{h\pm\frac{1}{2}}$ and $\tilde{k}_{k\pm\frac{1}{2}}$ are constant and all other terms are functions of the primary variables $[N_{n_c+1}, \ldots, N_{n_c+n_a}, P, N_w]$. Note that $N_w = N_{n_c+n_a+1}$.

5.2.2.3 Jacobian Matrix for the Matrix Media

The matrix governing equations are the volume constraint equation (Eq. 4.53) and the discrete forms of the matrix material balance equations (Eqs. 5.47 and 5.48). The volume constraint equation is only a function of the matrix primary variables ($\mathbf{x}_m$). The Jacobian terms of this equation are the same as those for the fracture system.

The partial derivatives of the material balance equations are functions of both fracture ($\mathbf{x}_f$) and matrix ($\mathbf{x}_m$) primary variables. For more details refer to Section 5.1.2.3 on page 96.

$\bar{\bar{D}}_{mm}$, Jacobian of the matrix equations with respect to the matrix independent variables, is a block square matrix of the size $N_b \times N_b$ where N_b is the number of matrix subdomains within a fracture gridblock. Each block in $\bar{\bar{D}}_{mm}$ is a $(n_c + n_a + 2) \times (n_c + n_a + 2)$ matrix. It includes derivatives of all matrix governing equations with respect to the matrix primary variables. $\bar{\bar{D}}_{mm}$ has the same structure of $\bar{\bar{A}}_{ff}$ (Eq. 5.45).

In Eq. 4.84, $\bar{\bar{C}}_{mf}$, Jacobian of the matrix equations with respect to the fracture independent variables, is a matrix of the size $N_b(n_c + n_a + 2) \times (n_c + n_a + 2)$.

$$\begin{aligned} \left[\bar{\bar{C}}_{mf}\right]_m &= \frac{\partial R^m_{conv}}{\partial \mathbf{x}_f} \\ \left[\bar{\bar{C}}_{mf}\right]_w &= \frac{\partial R^w_{conv}}{\partial \mathbf{x}_f} \end{aligned} \tag{5.37}$$

Note that the only non-zero entries in $\bar{\bar{C}}_{mf}$ come from the derivatives of residual equations of subdomains that have a face exposed to the fracture.
In the matrix form, $\bar{\bar{C}}_{mf}$ is expressed as

$$\bar{\bar{C}}_{mf} = \begin{bmatrix} \dfrac{\partial \mathbf{R}_1}{\partial \mathbf{x}_f} \\ \dfrac{\partial \mathbf{R}_2}{\partial \mathbf{x}_f} \\ \vdots \\ \dfrac{\partial \mathbf{R}_{N_b}}{\partial \mathbf{x}_f} \end{bmatrix} \tag{5.49}$$

where $\mathbf{R}_i = \left[R^v_i, R^m_{i1}, \ldots, R^m_{in_c+n_a}, R^w_i\right]^T$.

5.3 Matrix-Fracture Transfer Function

The matrix-fracture transfer term is the sum of fluid exchange over all faces of matrix subdomains that are in contact with the fracture. For a single matrix block (Fig. 5.1) it is defined as

$$
\begin{aligned}
&\tau_{mfi} = \\
&\sum_{N=1,M=1}^{N=1,M=M_{sub}} T_{mfi,M,N,vert}\left[(P_f - P_{m,M,N}) + (P_{c1jf} - P_{c1j,M,N}) - \gamma^* (D_f - D_{M,N})\right] \\
+ &\sum_{N=N_{sub},M=1}^{N=N_{sub},M=M_{sub}} T_{mfi,M,N,vert}\left[(P_f - P_{m,M,N}) + (P_{c1jf} - P_{c1j,M,N}) - \gamma^* (D_f - D_{M,N})\right] \\
+ &\sum_{N=1,M=M_{sub}}^{N=N_{sub},M=M_{sub}} T_{mfi,M,N,horz}\left[(P_f - P_{m,M,N}) + (P_{c1jf} - P_{c1j,M,N})\right]
\end{aligned}
\tag{5.50}
$$

The first term in the right-hand side of Eq. 5.50 accounts for the fluid transfer between the fracture and the matrix through the top faces of matrix subdomains, the second term accounts for the fluid transfer of fluid through bottom faces of matrix subdomains, and the last term accounts for the fluid exchange between matrix and fracture through the faces of the outermost rings.

The subdomain geometrical characteristics can be used to calculate the subdomain face transmissibilities. The top and bottom transmissibilities for component i in phase j are defined as

$$T_{mafi,M,N,vert} = \Gamma_{vert,M,N}\left[\omega\left(\frac{\xi_j}{\mu_j} x_{ij} krj\right)_m + (1-\omega)\left(\frac{\xi_j}{\mu_j} x_{ij} krj\right)_f\right] \tag{5.51}$$

where the geometrical constant , Γ_{vert}, is

$$\Gamma_{vert,M,N} = \frac{(L_{x,M} \times L_{y,M} - L_{x,M-1} \times L_{y,M-1})\, k_{mz}}{0.5 \times h_{sub,N}} \tag{5.52}$$

where L_x and L_y are the subdomain length in the x and y directions respectively and h_{sub} is the thickness of subdomain. The subdomian horizontal transmissibility is

$$T_{mafi,M,N,horz} = \Gamma_{horz,M,N} \left[\omega \left(\frac{\xi_j}{\mu_j} x_{ij} krj \right)_m + (1-\omega) \left(\frac{\xi_j}{\mu_j} x_{ij} krj \right)_f \right] \tag{5.53}$$

where

$$\Gamma_{horz,M,N} = 8 \times h_{sub,N} \left(\frac{L_{y,M} \times k_{xm}}{L_{x,M+1} - L_{x,M}} + \frac{L_{x,M} \times k_{ym}}{L_{y,M+1} - L_{y,M}} \right) \tag{5.54}$$

The geometrical constant contains the cross-sectional area to the flow, the absolute permeabilities and the distance between subdomain centers (Beckner *et al.*, 1991).

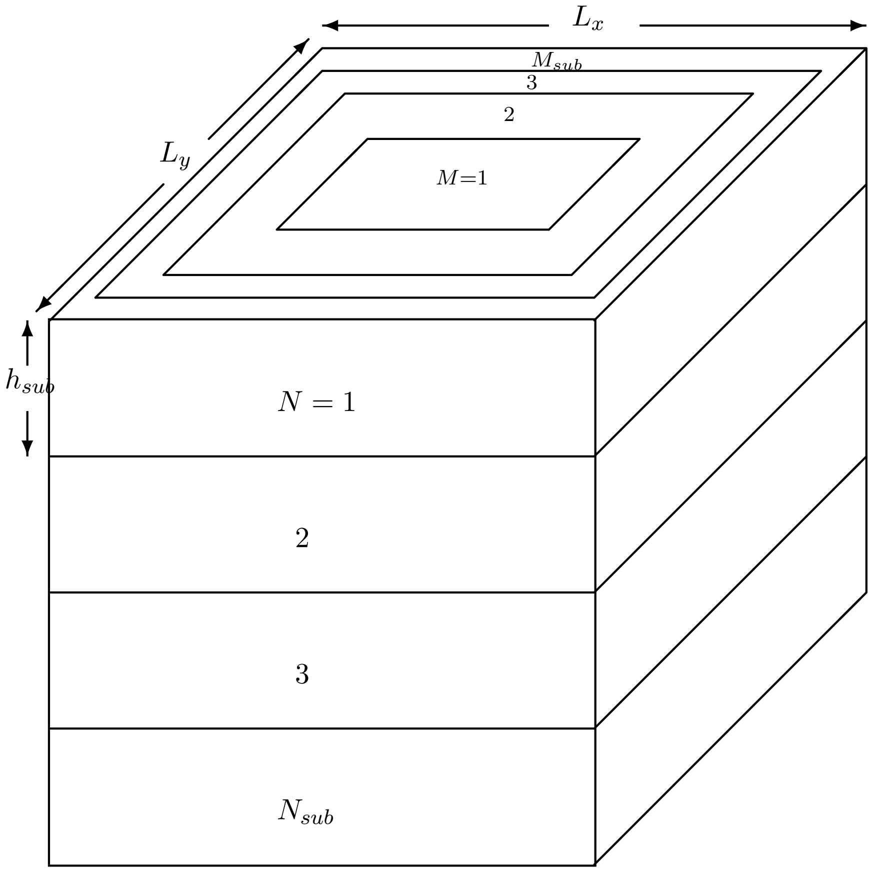


Figure 5.1: Multiple Interacting Media (MINC) Subgrids

Chapter 6

Full Tensor Formulations

6.1 Background

The presence of extensive networks of natural fractures creates a number of challenges to develop a reliable and accurate characterization of flow in fractured systems. Simulators need to somehow account for the complex geometry of the fractures and to incorporate all the relevant physics. Many methods are being developed to approximate the spacial distributions of rock properties such as porosity and permeability from the fine scale properties. Dual porosity and dual permeability models are widely used to provide basis for modeling naturally fractured reservoirs. The utilizations of these models rely on a) sufficiently accurate representation of the fluid exchange between the matrix and fracture, and b) precise representation of the fracture system as an equivalent porous media.

The advantage of dual porosity models is their ability to simulate complex local physical phenomena, *e.g.*, transport of fluid between the matrix and fracture. Use of the subgridding techniques to model transient flow within the matrix block considerably enhances the determination of the rate of the flow between the matrix and fracture. Notwithstanding, the ability of the dual porosity model is to simulate many complex problems by eliminating the length scale problem. The simple representation of the fracture system in dual porosity models is a huge obstacle. In the continuum models, it is assumed that the fracture system has a uniform fracture aperture, spacing and intensity. This simplification of the fracture system in the dual porosity models is widely considered to be inadequate to describe flow in fracture networks of complex connectivity and heterogeneity.

The formulation of fluid flow in most of the commercial simulators is based on the diagonal permeability tensor. The diagonal representation of permeability, which is customarily used in a dual porosity model, is valid only for the cases in which fractures are parallel to one of principal axes. This assumption cannot adequately describe flow characteristics where there is variation in fracture spacing, length, and orientation.

The discrete model is appealing from a modeling point of view, but the huge computational demand and burden in porting the fractures into the computational grid are its shortcomings. To incorporate the characteristics of the fracture system with a flow simulator, Lough *et al.* (1996) proposed a method to use a single porosity model with a full permeability tensor. They developed a method to transform the fractures into an equivalent permeability tensor. Then, the equivalent permeability tensor was ported to a conventional reservoir simulator to simulate the fractured reservoir. Both the fracture and matrix permeabilities were used in their model. A two-dimensional flow was assumed in fractures.

Full permeability tensor arises in several circumstances:

- When the principal directions of an anisotropic permeability are not aligned with the simulation coordinate directions.
- Non-orthogonal grid formulation.
- Scale up of permeability generate full permeability tensors even the fine-scale grids are isotropic and orthogonal.

- Fractured systems

Under these circumstances the effect of the off-diagonal terms of the permeability tensor has a strong effect on the fluid flow calculations. Therefore, a full permeability tensor representation is required in the formulation of flow simulators.

6.2 Full Tensor Representation of Flow

Darcy's law was originally developed based on a one-dimensional flow. Because directional properties have no effect on one-dimensional flow, permeability was considered as a scalar quantity. Collins (1961) and others have extended the Darcy's law in three dimensions by introducing a symmetric permeability tensor which can be transformed by a rotation of coordinate system. In a two-dimensional $x - y$ coordinate system, $\bar{\bar{\mathbf{k}}}$ is represented as

$$\bar{\bar{\mathbf{k}}} = \begin{pmatrix} k_{xx} & k_{xy} \\ k_{yx} & k_{yy} \end{pmatrix} \tag{6.1}$$

From a macroscopic point of view $\bar{\bar{\mathbf{k}}}$ is a symmetric positive definite matrix. The symmetry implies that $k_{xy} = k_{yx}$ and the positive definite implies that $k_{xx}k_{zz} \geq k_{xy}^2, k_{xx} > 0, k_{zz} > 0$ (Bear, 1972). These two characteristics are required for the permeability tensor to have a physical meaning. The eigenvalues of the permeability (*i.e.*, principal values) will be real only if the permeability is symmetric. The positive definite attribute of the permeability guarantees that the energy is always dissipated during flow, otherwise the flow will be against the pressure gradient (Durlofsky, 1991).

The directional permeability is based on the fact that the flow and the gradient are not required to be in the same direction. Thus, the permeability can be measured in either the direction of the flow or in the direction of the gradient.

This is the fundamental basis in the calculation of the permeability tensor in a homogeneous anisotropic medium (Long *et al.*, 1982).

Snow (1969) developed a mathematical model to calculate the full permeability tensor from a single extensive fracture of arbitrary orientation and aperture. The permeability tensor for a network of fractures is then formed by adding the individual permeability tensor of each individual fracture. In his model, all fractures are assumed to be of infinite extent.

Kasap and Lake (1989) developed an analytical method to calculate the effective permeability with the ability of handling tensor permeabilities. However, their method does not provide a general algorithm for the numerical calculation of effective or equivalent permeability.

Major challenges in calculating the equivalent permeability tensor are that 1) the boundary conditions imposed on the fine scale problem has significant impact on the equivalent permeability, and 2) an equivalent permeability which gives the correct flux may not accurately determine the gradient. White and Horne (1987) presented a numerical technique for calculating full permeability tensor from different boundary conditions. Their method, however, does not always result in a symmetric permeability tensor. Also, the discretization of their flow simulator was not based on the continuity of the flux across boundaries of control volumes.

Edwards (1995) developed a new method based on the continuity of the flux and pressure at the boundaries. In Edward's method, pressures were located at the cell vertices while the permeability was located at the cell center. This method was applicable to non-orthogonal or curvilinear grid systems. The difficulties arise in the implementation of this method in an existing simulator due to the fact that pressure is defined at cell vertices rather than the cell centers.

The scheme can be cell vertex or cell centered. If the pressure values are calculated at the cell vertices, then permeabilities are naturally placed at the cell centers. If the pressure values are calculated at the cell centers, then permeabilities have to be defined at the cell vertices. Arbogast *et al.* (1995) presented a cell-centered full tensor discretization scheme for a curvilinear grid system. They used a mixed finite element method because the method locally satisfies the mass conservation.

Lee *et al.* (1997) developed a simulator based on the continuity of flux at boundaries using a full tensor pressure equation. Both pressure and permeability were located at the center of gridblocks. They developed an explicit method for calculation of gridblock transmissibilities in Cartesian coordinate such that it can be incorporated in an existing finite difference reservoir simulator without difficulties. Theis method produces a cell molecule involving 27 gridblocks.

Aavatsmark *et al.* (1998) developed a 3D simulator based on the corner-point formulation using a full permeability tensor representation. To handle the nonorthogonal grid system, Aavatsmark *et al.* (1998) transformed the real-space problem to orthogonal Cartesian grid using a generalized coordinate transform. Lee *et al.* (1999) adopted Aavatsmark's method in an existing simulator without operating in transformed space. Their method produced a 27-point stencil finite-difference formula.

With the recent improvement in linear solver technologies, the use of unstructured grids have become widely popular. Edwards (2002) developed an unstructured flow simulator in conjunction with a full permeability tensor scheme.

All of the methods above implemented the full permeability tensor in a conventional single porosity simulator. The developed simulator can be used to simulate naturally fractured reservoirs using an effective or equivalent permeability tensor. It is certain that modeling a massively naturally fractured reservoir using a conventional single porosity model with full tensor no longer inherit the ability of simulating the complex local physical phenomena between the matrix and fracture systems.

Li (2001) pioneered work in the effort to handle the full permeability tensor in a dual porosity model. He adopted Aavatsmark *et al.* (1998) and Lee *et al.* (1999) methods in the UTCHEM chemical flooding simulator. In his implementation, he used the full permeability tensor scheme in the fracture system of the dual porosity model.

6.3 Full Tensor Implementation

Dealing with permeability anisotropy generally implies a flux approximation involving more than two gridblocks, as applies in corner point formulation. For two-point flux approximation the harmonic average of permeability is used to calculate the transmissibility, but there is no general form for multipoint flux approximation (MPFA). MPFA methods are classified based on the continuity conditions impose at the flux surfaces. Two subclasses of MPFA are O-method and U-method. In the O-method (Fig 6.1), a flux continuity and a single potential continuity are applied at each interface. In the U-method (Fig 6.1), the minimum number of surfaces are constrained. A flux continuity and a single potential continuity are applied at the central interface. The transmissibility is calculated at this interface. O-method yields a 27-point-stencil scheme while the U-method results in a 19-point stencil scheme (Aavatsmark *et al.*, 1998). The method used in this work is similar to the U-method which produces a cell molecule with 19-point stencil.

Original GPAS implementation is based on a diagonal permeability tensor. The mass balance equation (Eq. 4.6) for component i can be written as (Marcondes et al., 2005)

$$\begin{aligned}\frac{\partial(\phi_i N_i)}{\partial t} - \frac{\partial}{\partial x}\left(\sum_{j=1}^{n_p} \xi_j x_{ij} k_{11} \frac{\partial \Phi_j}{\partial x}\right) &- \frac{\partial}{\partial y}\left(\sum_{j=1}^{n_p} \xi_j x_{ij} k_{22} \frac{\partial \Phi_j}{\partial y}\right) \\ &- \frac{\partial}{\partial z}\left(\sum_{j=1}^{n_p} \xi_j x_{ij} k_{33} \frac{\partial \Phi_j}{\partial z}\right) - q_i + \tau_{mfi} = 0\end{aligned} \tag{6.2}$$

where Φ_j is the potential of phase j given by

$$\Phi_j = P_j - \gamma_j D \tag{6.3}$$

Our task here is to transform our problem to a problem similar to Eq. 6.2 which can be solved in a regular hexahedron domain, usually called computational domain.

The flux term for phase j in Eq. 6.2 can be written in a general form as following:

$$\overrightarrow{\nabla} \cdot \left(\varsigma_{ij} \bar{\bar{k}} \cdot \nabla \Phi_j\right) \tag{6.4}$$

where $\varsigma_{ij} = \xi_j x_{ij} \lambda_j$. $\bar{\bar{k}}$ is the permeability tensor and $\nabla \Phi_j$ is the pressure gradient given as

$$\bar{\bar{k}} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \qquad \nabla \Phi_j = \begin{bmatrix} \frac{\partial \Phi_j}{\partial x} \\ \frac{\partial \Phi_j}{\partial y} \\ \frac{\partial \Phi_j}{\partial z} \end{bmatrix} \tag{6.5}$$

The method here is defined by mapping the original grid system to an orthogonal grid system which can be considered as the computational grid system. Hence, the existing code with modified transmissibilities can be used to simulate the problem. Figure 6.2 shows a physical and computational domains for a 2D case.

For simplicity we develop the method based on a single phase flow flux term which can be easily extended to a multiphase flow by introducing the phase mobility (ς_{ij}) into equations. See Appendix C for multi-phase multi-component formulation. The single phase flow flux term can be generalized as follows:

$$\overrightarrow{\nabla} \cdot \left(\bar{\bar{k}} \cdot \nabla \Phi \right) \tag{6.6}$$

Expanding Eq. 6.6 gives

$$\begin{aligned}
\overrightarrow{\nabla} \cdot \left(\bar{\bar{k}} \cdot \nabla \Phi \right) = & \frac{\partial}{\partial x}\left(k_{11}\frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial x}\left(k_{12}\frac{\partial \Phi}{\partial y}\right) + \frac{\partial}{\partial x}\left(k_{13}\frac{\partial \Phi}{\partial z}\right) \\
& + \frac{\partial}{\partial y}\left(k_{21}\frac{\partial \Phi}{\partial y}\right) + \frac{\partial}{\partial y}\left(k_{22}\frac{\partial \Phi}{\partial y}\right) + \frac{\partial}{\partial y}\left(k_{23}\frac{\partial \Phi}{\partial z}\right) \\
& + \frac{\partial}{\partial z}\left(k_{31}\frac{\partial \Phi}{\partial x}\right) + \frac{\partial}{\partial z}\left(k_{32}\frac{\partial \Phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_{33}\frac{\partial \Phi}{\partial z}\right) \\
= & \frac{\partial}{\partial x}\left(k_{11}\frac{\partial \Phi}{\partial x} + k_{12}\frac{\partial \Phi}{\partial y} + k_{13}\frac{\partial \Phi}{\partial z}\right) \\
& + \frac{\partial}{\partial y}\left(k_{21}\frac{\partial \Phi}{\partial x} + k_{22}\frac{\partial \Phi}{\partial y} + k_{23}\frac{\partial \Phi}{\partial z}\right) \\
& + \frac{\partial}{\partial z}\left(k_{31}\frac{\partial \Phi}{\partial x} + k_{32}\frac{\partial \Phi}{\partial y} + k_{33}\frac{\partial \Phi}{\partial z}\right) \\
= & \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z}
\end{aligned} \tag{6.7}$$

Let (ξ, η, γ) be a right-hand Cartesian coordinate system and (x, y, z) be a curvilinear coordinate system. Assume there is a one-to-one mapping between the position vector $r = [x, y, z]^T$ and $\varrho = [\xi, \eta, \gamma]^T$ then

$$\xi = \xi(x, y, z); \quad \eta = \eta(x, y, z); \quad \gamma = \gamma(x, y, z) \tag{6.8}$$

We need to transform everything from (x, y, z) coordinate system to (ξ, η, γ) coordinate system. Applying the chain rule, the gradient of E, F, and G in $x, y,$

and z directions are given as follows:

$$\begin{aligned}
\frac{\partial E}{\partial x} &= \frac{\partial E}{\partial \xi}\frac{d\xi}{dx} + \frac{\partial E}{\partial \eta}\frac{d\eta}{dx} + \frac{\partial E}{\partial \gamma}\frac{d\gamma}{dx} \\
\frac{\partial F}{\partial y} &= \frac{\partial F}{\partial \xi}\frac{d\xi}{dy} + \frac{\partial F}{\partial \eta}\frac{d\eta}{dy} + \frac{\partial F}{\partial \gamma}\frac{d\gamma}{dy} \\
\frac{\partial G}{\partial z} &= \frac{\partial G}{\partial \xi}\frac{d\xi}{dz} + \frac{\partial G}{\partial \eta}\frac{d\eta}{dz} + \frac{\partial G}{\partial \gamma}\frac{d\gamma}{dz}
\end{aligned} \tag{6.9}$$

The Jacobian matrix of transformation $\mathbf{J}$ is defined as

$$\mathbf{J} = \left|\frac{dr}{d\varrho}\right| \tag{6.10}$$

Dividing each term of Eq. 6.9 by $\mathbf{J}$ and replacing the results in Eq. 6.7 and adding and subtracting terms of type $E\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{\mathbf{J}}\right), F\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{\mathbf{J}}\right), \ldots$ will give

$$\begin{aligned}
\overrightarrow{\nabla}\cdot\left(\bar{\bar{k}}\cdot\nabla\Phi\right) = {} & \frac{\partial}{\partial \xi}\left(\frac{\xi_x E + \xi_y F + \xi_z G}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_x E + \eta_y F + \eta_z G}{\mathbf{J}}\right) \\
& + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_x E + \gamma_y F + \gamma_z G}{\mathbf{J}}\right) \\
& - E\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_x}{\mathbf{J}}\right) + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_x}{\mathbf{J}}\right)\right] \\
& - F\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_y}{\mathbf{J}}\right) + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_z}{\mathbf{J}}\right)\right] \\
& - G\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_z}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_z}{\mathbf{J}}\right) + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_z}{\mathbf{J}}\right)\right]
\end{aligned} \tag{6.11}$$

where $\xi_x = \frac{d\xi}{dx}, \xi_y = \frac{d\xi}{dy}, \ldots.,$ and so on. It can be shown that each term in the brackets is equal to zero. Thus, Eq. 6.11 can be written as

$$\begin{aligned}
\overrightarrow{\nabla}\cdot\left(\bar{\bar{k}}\cdot\nabla\Phi\right) = {} & \frac{\partial}{\partial \xi}\left(\frac{\xi_x E + \xi_y F + \xi_z G}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_x E + \eta_y F + \eta_z G}{\mathbf{J}}\right) \\
& + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_x E + \gamma_y F + \gamma_z G}{\mathbf{J}}\right)
\end{aligned} \tag{6.12}$$

Now we use the chain rule to calculate derivatives of E, F, and G (see Eq. 6.7 for the definition of E, F, and G) yields

$$\begin{aligned} E = k_{11}\frac{\partial \Phi}{\partial x} + k_{12}\frac{\partial \Phi}{\partial y} + k_{13}\frac{\partial \Phi}{\partial z} = \quad & k_{11}\left(\frac{\partial \Phi}{\partial \xi}\xi_x + \frac{\partial \Phi}{\partial \eta}\eta_x + \frac{\partial \Phi}{\partial \gamma}\gamma_x\right) \\ & + k_{12}\left(\frac{\partial \Phi}{\partial \xi}\xi_y + \frac{\partial \Phi}{\partial \eta}\eta_y + \frac{\partial \Phi}{\partial \gamma}\gamma_y\right) \\ & + k_{13}\left(\frac{\partial \Phi}{\partial \xi}\xi_z + \frac{\partial \Phi}{\partial \eta}\eta_z + \frac{\partial \Phi}{\partial \gamma}\gamma_z\right) \end{aligned} \tag{6.13}$$

$$\begin{aligned} F = k_{21}\frac{\partial \Phi}{\partial x} + k_{22}\frac{\partial \Phi}{\partial y} + k_{23}\frac{\partial \Phi}{\partial z} = \quad & k_{21}\left(\frac{\partial \Phi}{\partial \xi}\xi_x + \frac{\partial \Phi}{\partial \eta}\eta_x + \frac{\partial \Phi}{\partial \gamma}\gamma_x\right) \\ & + k_{22}\left(\frac{\partial \Phi}{\partial \xi}\xi_y + \frac{\partial \Phi}{\partial \eta}\eta_y + \frac{\partial \Phi}{\partial \gamma}\gamma_y\right) \\ & + k_{23}\left(\frac{\partial \Phi}{\partial \xi}\xi_z + \frac{\partial \Phi}{\partial \eta}\eta_z + \frac{\partial \Phi}{\partial \gamma}\gamma_z\right) \end{aligned} \tag{6.14}$$

$$\begin{aligned} G = k_{31}\frac{\partial \Phi}{\partial x} + k_{32}\frac{\partial \Phi}{\partial y} + k_{33}\frac{\partial \Phi}{\partial z} = \quad & k_{31}\left(\frac{\partial \Phi}{\partial \xi}\xi_x + \frac{\partial \Phi}{\partial \eta}\eta_x + \frac{\partial \Phi}{\partial \gamma}\gamma_x\right) \\ & + k_{32}\left(\frac{\partial \Phi}{\partial \xi}\xi_y + \frac{\partial \Phi}{\partial \eta}\eta_y + \frac{\partial \Phi}{\partial \gamma}\gamma_y\right) \\ & + k_{33}\left(\frac{\partial \Phi}{\partial \xi}\xi_z + \frac{\partial \Phi}{\partial \eta}\eta_z + \frac{\partial \Phi}{\partial \gamma}\gamma_z\right) \end{aligned} \tag{6.15}$$

Replacing E, F, and G in Eq. 6.12, we obtain

$$
\begin{aligned}
&\overrightarrow{\nabla}\cdot\left(\bar{\bar{k}}\cdot\nabla\Phi\right)=\\
&\frac{\partial}{\partial\xi}\Bigg\{\frac{\xi_x}{\mathbf{J}}\Bigg[k_{11}\left(\frac{\partial\Phi}{\partial\xi}\xi_x+\frac{\partial\Phi}{\partial\eta}\eta_x+\frac{\partial\Phi}{\partial\gamma}\gamma_x\right)+k_{12}\left(\frac{\partial\Phi}{\partial\xi}\xi_y+\frac{\partial\Phi}{\partial\eta}\eta_y+\frac{\partial\Phi}{\partial\gamma}\gamma_y\right)\\
&\qquad\qquad+k_{13}\left(\frac{\partial\Phi}{\partial\xi}\xi_z+\frac{\partial\Phi}{\partial\eta}\eta_z+\frac{\partial\Phi}{\partial\gamma}\gamma_z\right)\Bigg]\\
&\qquad\frac{\xi_y}{\mathbf{J}}\Bigg[k_{21}\left(\frac{\partial\Phi}{\partial\xi}\xi_x+\frac{\partial\Phi}{\partial\eta}\eta_x+\frac{\partial\Phi}{\partial\gamma}\gamma_x\right)+k_{22}\left(\frac{\partial\Phi}{\partial\xi}\xi_y+\frac{\partial\Phi}{\partial\eta}\eta_y+\frac{\partial\Phi}{\partial\gamma}\gamma_y\right)\\
&\qquad\qquad+k_{23}\left(\frac{\partial\Phi}{\partial\xi}\xi_z+\frac{\partial\Phi}{\partial\eta}\eta_z+\frac{\partial\Phi}{\partial\gamma}\gamma_z\right)\Bigg]\\
&\qquad\frac{\xi_z}{\mathbf{J}}\Bigg[k_{31}\left(\frac{\partial\Phi}{\partial\xi}\xi_x+\frac{\partial\Phi}{\partial\eta}\eta_x+\frac{\partial\Phi}{\partial\gamma}\gamma_x\right)+k_{32}\left(\frac{\partial\Phi}{\partial\xi}\xi_y+\frac{\partial\Phi}{\partial\eta}\eta_y+\frac{\partial\Phi}{\partial\gamma}\gamma_y\right)\\
&\qquad\qquad+k_{33}\left(\frac{\partial\Phi}{\partial\xi}\xi_z+\frac{\partial\Phi}{\partial\eta}\eta_z+\frac{\partial\Phi}{\partial\gamma}\gamma_z\right)\Bigg]\Bigg\}\\
&+\frac{\partial}{\partial\eta}\Bigg\{\frac{\eta_x}{\mathbf{J}}\Bigg[k_{11}\left(\frac{\partial\Phi}{\partial\xi}\xi_x+\frac{\partial\Phi}{\partial\eta}\eta_x+\frac{\partial\Phi}{\partial\gamma}\gamma_x\right)+k_{12}\left(\frac{\partial\Phi}{\partial\xi}\xi_y+\frac{\partial\Phi}{\partial\eta}\eta_y+\frac{\partial\Phi}{\partial\gamma}\gamma_y\right)\\
&\qquad\qquad+k_{13}\left(\frac{\partial\Phi}{\partial\xi}\xi_z+\frac{\partial\Phi}{\partial\eta}\eta_z+\frac{\partial\Phi}{\partial\gamma}\gamma_z\right)\Bigg]\\
&\qquad\frac{\eta_y}{\mathbf{J}}\Bigg[k_{21}\left(\frac{\partial\Phi}{\partial\xi}\xi_x+\frac{\partial\Phi}{\partial\eta}\eta_x+\frac{\partial\Phi}{\partial\gamma}\gamma_x\right)+k_{22}\left(\frac{\partial\Phi}{\partial\xi}\xi_y+\frac{\partial\Phi}{\partial\eta}\eta_y+\frac{\partial\Phi}{\partial\gamma}\gamma_y\right)\\
&\qquad\qquad+k_{23}\left(\frac{\partial\Phi}{\partial\xi}\xi_z+\frac{\partial\Phi}{\partial\eta}\eta_z+\frac{\partial\Phi}{\partial\gamma}\gamma_z\right)\Bigg]\\
&\qquad\frac{\eta_z}{\mathbf{J}}\Bigg[k_{31}\left(\frac{\partial\Phi}{\partial\xi}\xi_x+\frac{\partial\Phi}{\partial\eta}\eta_x+\frac{\partial\Phi}{\partial\gamma}\gamma_x\right)+k_{32}\left(\frac{\partial\Phi}{\partial\xi}\xi_y+\frac{\partial\Phi}{\partial\eta}\eta_y+\frac{\partial\Phi}{\partial\gamma}\gamma_y\right)\\
&\qquad\qquad+k_{33}\left(\frac{\partial\Phi}{\partial\xi}\xi_z+\frac{\partial\Phi}{\partial\eta}\eta_z+\frac{\partial\Phi}{\partial\gamma}\gamma_z\right)\Bigg]\Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
+\frac{\partial}{\partial \gamma}\Bigg\{ & \frac{\gamma_x}{\mathbf{J}}\Bigg[k_{11}\left(\frac{\partial \Phi}{\partial \xi}\xi_x+\frac{\partial \Phi}{\partial \eta}\eta_x+\frac{\partial \Phi}{\partial \gamma}\gamma_x\right)+k_{12}\left(\frac{\partial \Phi}{\partial \xi}\xi_y+\frac{\partial \Phi}{\partial \eta}\eta_y+\frac{\partial \Phi}{\partial \gamma}\gamma_y\right) \\
& \quad +k_{13}\left(\frac{\partial \Phi}{\partial \xi}\xi_z+\frac{\partial \Phi}{\partial \eta}\eta_z+\frac{\partial \Phi}{\partial \gamma}\gamma_z\right)\Bigg] \\
& \frac{\gamma_y}{\mathbf{J}}\Bigg[k_{21}\left(\frac{\partial \Phi}{\partial \xi}\xi_x+\frac{\partial \Phi}{\partial \eta}\eta_x+\frac{\partial \Phi}{\partial \gamma}\gamma_x\right)+k_{22}\left(\frac{\partial \Phi}{\partial \xi}\xi_y+\frac{\partial \Phi}{\partial \eta}\eta_y+\frac{\partial \Phi}{\partial \gamma}\gamma_y\right) \\
& \quad +k_{23}\left(\frac{\partial \Phi}{\partial \xi}\xi_z+\frac{\partial \Phi}{\partial \eta}\eta_z+\frac{\partial \Phi}{\partial \gamma}\gamma_z\right)\Bigg] \\
& \frac{\gamma_z}{\mathbf{J}}\Bigg[k_{31}\left(\frac{\partial \Phi}{\partial \xi}\xi_x+\frac{\partial \Phi}{\partial \eta}\eta_x+\frac{\partial \Phi}{\partial \gamma}\gamma_x\right)+k_{32}\left(\frac{\partial \Phi}{\partial \xi}\xi_y+\frac{\partial \Phi}{\partial \eta}\eta_y+\frac{\partial \Phi}{\partial \gamma}\gamma_y\right) \\
& \quad +k_{33}\left(\frac{\partial \Phi}{\partial \xi}\xi_z+\frac{\partial \Phi}{\partial \eta}\eta_z+\frac{\partial \Phi}{\partial \gamma}\gamma_z\right)\Bigg]\Bigg\}
\end{aligned}
\tag{6.16}
$$

By factoring similar terms and manipulating the above equation we have

$$
\begin{aligned}
&\overrightarrow{\nabla} \cdot \left(\bar{\bar{k}} \cdot \nabla\Phi\right) = \\
&\quad \frac{\partial}{\partial \xi} \left\{ \left[\frac{1}{\mathbf{J}} \left(k_{11}\xi_x^2 + k_{22}\xi_y^2 + k_{33}\xi_z^2\right) \right.\right. \\
&\qquad \left.\left. + \frac{1}{\mathbf{J}} \left(k_{12}\xi_x\xi_y + k_{13}\xi_x\xi_z + k_{21}\xi_x\xi_y + k_{23}\xi_y\xi_z + k_{31}\xi_z\xi_x + k_{32}\xi_z\xi_y\right) \right] \frac{\partial\Phi}{\partial \xi} \right\} \\
&\quad + \frac{\partial}{\partial \eta} \left\{ \left[\frac{1}{\mathbf{J}} \left(k_{11}\eta_x^2 + k_{22}\eta_y^2 + k_{33}\eta_z^2\right) \right.\right. \\
&\qquad \left.\left. + \frac{1}{\mathbf{J}} \left(k_{12}\eta_x\eta_y + k_{13}\eta_x\eta_z + k_{21}\eta_x\eta_y + k_{23}\eta_y\eta_z + k_{31}\eta_z\eta_x + k_{32}\eta_z\eta_y\right) \right] \frac{\partial\Phi}{\partial \eta} \right\} \\
&\quad + \frac{\partial}{\partial \gamma} \left\{ \left[\frac{1}{\mathbf{J}} \left(k_{11}\gamma_x^2 + k_{22}\gamma_y^2 + k_{33}\gamma_z^2\right) \right.\right. \\
&\qquad \left.\left. + \frac{1}{\mathbf{J}} \left(k_{12}\gamma_x\gamma_y + k_{13}\gamma_x\gamma_z + k_{21}\gamma_x\gamma_y + k_{23}\gamma_y\gamma_z + k_{31}\gamma_z\gamma_x + k_{32}\gamma_z\gamma_y\right) \right] \frac{\partial\Phi}{\partial \gamma} \right\} \\
&\quad + \frac{\partial}{\partial \xi} \left\{ \left[\frac{1}{\mathbf{J}} \left(k_{11}\xi_x\eta_x + k_{12}\xi_x\eta_y + k_{13}\xi_x\eta_z\right) + \frac{1}{\mathbf{J}} \left(k_{21}\xi_y\eta_x + k_{22}\xi_y\eta_y + k_{23}\xi_y\eta_z\right) \right.\right. \\
&\qquad \left.\left. + \frac{1}{\mathbf{J}} \left(k_{31}\xi_z\eta_x + k_{32}\xi_z\eta_y + k_{33}\xi_z\eta_z\right) \right] \frac{\partial\Phi}{\partial \eta} \right\} \\
&\quad + \frac{\partial}{\partial \xi} \left\{ \left[\frac{1}{\mathbf{J}} \left(k_{11}\xi_x\gamma_x + k_{12}\xi_x\gamma_y + k_{13}\xi_x\gamma_z\right) + \frac{1}{\mathbf{J}} \left(k_{21}\xi_y\gamma_x + k_{22}\xi_y\gamma_y + k_{23}\xi_y\gamma_z\right) \right.\right. \\
&\qquad \left.\left. + \frac{1}{\mathbf{J}} \left(k_{31}\xi_z\gamma_x + k_{32}\xi_z\gamma_y + k_{33}\xi_z\gamma_z\right) \right] \frac{\partial\Phi}{\partial \gamma} \right\}
\end{aligned}
$$

$$
\begin{aligned}
&+\frac{\partial}{\partial\eta}\left\{\left[\frac{1}{\mathbf{J}}\left(k_{11}\eta_x\xi_x+k_{12}\eta_x\xi_y+k_{13}\eta_x\xi_z\right)+\frac{1}{\mathbf{J}}\left(k_{21}\eta_y\xi_x+k_{22}\eta_y\xi_y+k_{23}\eta_y\xi_z\right)\right.\right.\\
&\qquad\left.\left.+\frac{1}{\mathbf{J}}\left(k_{31}\eta_z\xi_x+k_{32}\eta_z\xi_y+k_{33}\eta_z\xi_z\right)\right]\frac{\partial\Phi}{\partial\xi}\right\}\\
&+\frac{\partial}{\partial\eta}\left\{\left[\frac{1}{\mathbf{J}}\left(k_{11}\eta_x\gamma_x+k_{12}\eta_x\gamma_y+k_{13}\eta_x\gamma_z\right)+\frac{1}{\mathbf{J}}\left(k_{21}\eta_y\gamma_x+k_{22}\eta_y\gamma_y+k_{23}\eta_y\gamma_z\right)\right.\right.\\
&\qquad\left.\left.+\frac{1}{\mathbf{J}}\left(k_{31}\eta_z\gamma_x+k_{32}\eta_z\gamma_y+k_{33}\eta_z\gamma_z\right)\right]\frac{\partial\Phi}{\partial\gamma}\right\}\\
&+\frac{\partial}{\partial\gamma}\left\{\left[\frac{1}{\mathbf{J}}\left(k_{11}\gamma_x\xi_x+k_{12}\gamma_x\xi_y+k_{13}\gamma_x\xi_z\right)+\frac{1}{\mathbf{J}}\left(k_{21}\gamma_y\xi_x+k_{22}\gamma_y\xi_y+k_{23}\gamma_y\xi_z\right)\right.\right.\\
&\qquad\left.\left.+\frac{1}{\mathbf{J}}\left(k_{31}\gamma_z\xi_x+k_{32}\gamma_z\xi_y+k_{33}\gamma_z\xi_z\right)\right]\frac{\partial\Phi}{\partial\xi}\right\}\\
&+\frac{\partial}{\partial\gamma}\left\{\left[\frac{1}{\mathbf{J}}\left(k_{11}\gamma_x\eta_x+k_{12}\gamma_x\eta_y+k_{13}\gamma_x\eta_z\right)+\frac{1}{\mathbf{J}}\left(k_{21}\gamma_y\eta_x+k_{22}\gamma_y\eta_y+k_{23}\gamma_y\eta_z\right)\right.\right.\\
&\qquad\left.\left.+\frac{1}{\mathbf{J}}\left(k_{31}\gamma_z\eta_x+k_{32}\gamma_z\eta_y+k_{33}\gamma_z\eta_z\right)\right]\frac{\partial\Phi}{\partial\eta}\right\}
\end{aligned}
\tag{6.17}
$$

After mapping the original grid system (x, y, z) to the orthogonal computational grid system (ξ, η, γ) the new terms are

$$
\begin{aligned}
T_{11}=\frac{1}{\mathbf{J}}&\left(k_{11}\xi_x^2+k_{22}\xi_y^2+k_{33}\xi_z^2\right.\\
+&\left.\ k_{12}\xi_x\xi_y+k_{13}\xi_x\xi_z+k_{21}\xi_x\xi_y+k_{23}\xi_y\xi_z+k_{31}\xi_z\xi_x+k_{32}\xi_z\xi_y\right)
\end{aligned}
\tag{6.18}
$$

$$
\begin{aligned}
T_{22}=\frac{1}{\mathbf{J}}&\left(k_{11}\eta_x^2+k_{22}\eta_y^2+k_{33}\eta_z^2\right.\\
+&\left.\ k_{12}\eta_x\eta_y+k_{13}\eta_x\eta_z+k_{21}\eta_x\eta_y+k_{23}\eta_y\eta_z+k_{31}\eta_z\eta_x+k_{32}\eta_z\eta_y\right)
\end{aligned}
\tag{6.19}
$$

$$
\begin{aligned}
T_{33} = \frac{1}{\mathbf{J}} \big(& k_{11}\gamma_x^2 + k_{22}\gamma_y^2 + k_{33}\gamma_z^2 \\
+ \ & k_{12}\gamma_x\gamma_y + k_{13}\gamma_x\gamma_z + k_{21}\gamma_x\gamma_y + k_{23}\gamma_y\gamma_z + k_{31}\gamma_z\gamma_x + k_{32}\gamma_z\gamma_y \big)
\end{aligned}
\tag{6.20}
$$

$$
\begin{aligned}
T_{12} = T_{21} = \frac{1}{\mathbf{J}} (& k_{11}\xi_x\eta_x + k_{12}\xi_x\eta_y + k_{13}\xi_x\eta_z \\
+ \ & k_{21}\xi_y\eta_x + k_{22}\xi_y\eta_y + k_{23}\xi_y\eta_z + k_{31}\xi_z\eta_x + k_{32}\xi_z\eta_y + k_{33}\xi_z\eta_z)
\end{aligned}
\tag{6.21}
$$

$$
\begin{aligned}
T_{13} = T_{31} = \frac{1}{\mathbf{J}} (& k_{11}\xi_x\gamma_x + k_{12}\xi_x\gamma_y + k_{13}\xi_x\gamma_z \\
+ \ & k_{21}\xi_y\gamma_x + k_{22}\xi_y\gamma_y + k_{23}\xi_y\gamma_z + k_{31}\xi_z\gamma_x + k_{32}\xi_z\gamma_y + k_{33}\xi_z\gamma_z)
\end{aligned}
\tag{6.22}
$$

$$
\begin{aligned}
T_{23} = T_{32} = \frac{1}{\mathbf{J}} (& k_{11}\eta_x\gamma_x + k_{12}\eta_x\gamma_y + k_{13}\eta_x\gamma_z \\
+ \ & k_{21}\eta_y\gamma_x + k_{22}\eta_y\gamma_y + k_{23}\eta_y\gamma_z + k_{31}\eta_z\gamma_x + k_{32}\eta_z\gamma_y + k_{33}\eta_z\gamma_z)
\end{aligned}
\tag{6.23}
$$

As shown in the above equations, the new transmissibility terms are symmetric.

6.4 Numerical Approximation

Integrating Eq. 6.17 in space (control volume of Fig. 6.3) and considering a fully implicit formulation for all terms except for the derivatives of the cross terms, the discrete form of Eq. 6.17 will be given as follow

$$\begin{aligned}
&\left\{T_{11e}\left(\frac{\partial\Phi}{\partial\xi}\right)_e \Delta\eta\Delta\gamma - T_{11w}\left(\frac{\partial\Phi}{\partial\xi}\right)_w \Delta\eta\Delta\gamma\right\}^{n+1} \\
+&\left\{T_{12e}\left(\frac{\partial\Phi}{\partial\eta}\right)_e \Delta\eta\Delta\gamma - T_{12w}\left(\frac{\partial\Phi}{\partial\eta}\right)_w \Delta\eta\Delta\gamma\right\}^{n} \\
+&\left\{T_{13e}\left(\frac{\partial\Phi}{\partial\gamma}\right)_e \Delta\eta\Delta\gamma - T_{13w}\left(\frac{\partial\Phi}{\partial\gamma}\right)_w \Delta\eta\Delta\gamma\right\}^{n} \\
+&\left\{T_{21n}\left(\frac{\partial\Phi}{\partial\xi}\right)_n \Delta\xi\Delta\gamma - T_{21s}\left(\frac{\partial\Phi}{\partial\xi}\right)_s \Delta\xi\Delta\gamma\right\}^{n} \\
+&\left\{T_{22n}\left(\frac{\partial\Phi}{\partial\eta}\right)_n \Delta\xi\Delta\gamma - T_{22s}\left(\frac{\partial\Phi}{\partial\eta}\right)_s \Delta\xi\Delta\gamma\right\}^{n+1} \\
+&\left\{T_{23n}\left(\frac{\partial\Phi}{\partial\gamma}\right)_n \Delta\xi\Delta\gamma - T_{23s}\left(\frac{\partial\Phi}{\partial\gamma}\right)_s \Delta\xi\Delta\gamma\right\}^{n} \\
+&\left\{T_{31f}\left(\frac{\partial\Phi}{\partial\xi}\right)_f \Delta\xi\Delta\eta - T_{31b}\left(\frac{\partial\Phi}{\partial\xi}\right)_b \Delta\xi\Delta\eta\right\}^{n} \\
+&\left\{T_{32f}\left(\frac{\partial\Phi}{\partial\eta}\right)_f \Delta\xi\Delta\eta - T_{32b}\left(\frac{\partial\Phi}{\partial\eta}\right)_b \Delta\xi\Delta\eta\right\}^{n} \\
+&\left\{T_{33f}\left(\frac{\partial\Phi}{\partial\gamma}\right)_f \Delta\xi\Delta\eta - T_{33b}\left(\frac{\partial\Phi}{\partial\gamma}\right)_b \Delta\xi\Delta\eta\right\}^{n+1}
\end{aligned} \tag{6.24}$$

The derivatives in Eq. 6.24 are evaluated using a central difference scheme. This method produces a cell molecule with 19 cell grids involved (Fig. 6.4). Using the cell molecule in Fig. 6.4, the derivatives at each face of control volume P are given by

$$\begin{aligned}
\left(\frac{\partial\Phi}{\partial\xi}\right)_e &= \frac{\Phi_E-\Phi_P}{\Delta\xi}; &\quad \left(\frac{\partial\Phi}{\partial\xi}\right)_w &= \frac{\Phi_P-\Phi_W}{\Delta\xi} \\
\left(\frac{\partial\Phi}{\partial\eta}\right)_n &= \frac{\Phi_N-\Phi_P}{\Delta\eta}; &\quad \left(\frac{\partial\Phi}{\partial\eta}\right)_s &= \frac{\Phi_P-\Phi_S}{\Delta\eta} \\
\left(\frac{\partial\Phi}{\partial\gamma}\right)_f &= \frac{\Phi_F-\Phi_P}{\Delta\xi}; &\quad \left(\frac{\partial\Phi}{\partial\gamma}\right)_b &= \frac{\Phi_P-\Phi_B}{\Delta\gamma}
\end{aligned} \tag{6.25}$$

$$
\begin{aligned}
\left(\frac{\partial \Phi}{\partial \eta}\right)_e &= \frac{\Phi_N + \Phi_{NE} - \Phi_S - \Phi_{SE}}{4\Delta\eta}; & \left(\frac{\partial \Phi}{\partial \eta}\right)_w &= \frac{\Phi_N + \Phi_{NW} - \Phi_S - \Phi_{SW}}{4\Delta\eta} \\
\left(\frac{\partial \Phi}{\partial \gamma}\right)_e &= \frac{\Phi_F + \Phi_{FE} - \Phi_B - \Phi_{BE}}{4\Delta\gamma}; & \left(\frac{\partial \Phi}{\partial \gamma}\right)_w &= \frac{\Phi_F + \Phi_{FW} - \Phi_B - \Phi_{BW}}{4\Delta\eta} \\
\left(\frac{\partial \Phi}{\partial \xi}\right)_n &= \frac{\Phi_E + \Phi_{NE} - \Phi_W - \Phi_{NW}}{4\Delta\xi}; & \left(\frac{\partial \Phi}{\partial \xi}\right)_s &= \frac{\Phi_E + \Phi_{SE} - \Phi_W - \Phi_{SW}}{4\Delta\xi} \\
\left(\frac{\partial \Phi}{\partial \gamma}\right)_n &= \frac{\Phi_F + \Phi_{FN} - \Phi_B - \Phi_{BN}}{4\Delta\gamma}; & \left(\frac{\partial \Phi}{\partial \gamma}\right)_s &= \frac{\Phi_F + \Phi_{FS} - \Phi_B - \Phi_{BS}}{4\Delta\gamma} \\
\left(\frac{\partial \Phi}{\partial \xi}\right)_f &= \frac{\Phi_E + \Phi_{FE} - \Phi_W - \Phi_{FW}}{4\Delta\xi}; & \left(\frac{\partial \Phi}{\partial \xi}\right)_b &= \frac{\Phi_E + \Phi_{BE} - \Phi_W - \Phi_{BW}}{4\Delta\xi} \\
\left(\frac{\partial \Phi}{\partial \eta}\right)_f &= \frac{\Phi_N + \Phi_{FN} - \Phi_S - \Phi_{FS}}{4\Delta\eta}; & \left(\frac{\partial \Phi}{\partial \eta}\right)_b &= \frac{\Phi_N + \Phi_{BN} - \Phi_S - \Phi_{BS}}{4\Delta\eta}
\end{aligned}
\tag{6.26}
$$

The above formulation can be extended to a multiphase and multicomponent system by using phase mobility and component mole fraction. Appendix C covers formulation of full tensor for multiphase and multicomponent systems.

6.5 Equivalent Fracture Permeability

The advantage of the dual porosity model is the ability to simulate complex recovery mechanisms with reasonable computational effort. Almost all of the dual porosity simulators use the diagonal permeability tensor for fracture systems. The diagonal permeability tensor may not represent highly heterogeneous systems.

Snow (1969) introduced a new method to calculate an equivalent permeability tensor for a fractured system. He assumed that there was no energy loss at the intersection of fractures and the flow along a fracture is proportional to the normal projection of gradient of potential. As the mutual interface was neglected, the total flow through the fractures was calculated by the vector sum of individual fracture.

He introduced the following equation for the equivalent permeability

$$k_{ij} = \frac{2}{3} \sum \frac{b^3}{|n_i D_i|} (\delta_{ij} - n_i n_j) \tag{6.27}$$

where b is the half aperture, n_i is the projection of the normal vector $\vec{n}$ in i direction, n_j is the projection of the normal vector $\vec{n}$ in j direction, D is the sampling length, and δ is the Kroneker delta.

Oda (1985) introduced a method to calculate the permeability tensor of the fractured reservoir from the fracture geometry. He expressed the fracture permeability as a function of a symmetric, second-rank tensor P_{ij} (called the fracture tensor) which only depends on the geometrical properties of fractures, such as aperture, size, and an orientation, through the following equation:

$$k_{ij} = \zeta (P_{kk} \delta_{ij} - P_{ij}) \tag{6.28}$$

where

$$P_{ij} = \frac{\pi \rho}{4} \int_0^\infty \int_0^\infty \int_\Omega r^2 t^3 n_i n_j E(\vec{n}, r, t) d\Omega dr dt \tag{6.29}$$

and

$$P_{kk} = P_{11} + P_{22} + P_{33} \tag{6.30}$$

where k_{ij} is the permeability tensor, ζ is a constant and is equal $\frac{1}{12}$ for infinitely extended fractures, ρ is fracture volume density, r is the fracture radius, t is the aperture, Ω is the angle around the fracture normal $\vec{n}$, and $E(\vec{n}, r, t)$ is a density function for fractures.

The notation P_{ij} is defined as the crack tensor which is only related to the crack geometry, *i.e.*, the crack shape, crack size, aperture and orientation. Figure

6.5 gives a schematic of a fractured studied by Oda (1985).

The advantage of Oda's method is that the calculation does not require the flow simulator, but its shortcoming is that it does not account for the fracture size and connectivity of fractures. Therefore, it is limited for a well connected fracture system.

In this study, the Oda's method is used to calculate the equivalent permeability by projecting the fracture permeability onto its plane, and then scaling it by the ratio of the fracture volume to the gridblock volume. Figure 6.5 shows a single fracture with aperture B, bounded by two infinite parallel plates. The slit's scalar permeability can be expressed by

$$k = \frac{B^2}{12} \tag{6.31}$$

Applying Darcy's law to a single fracture i gives

$$\vec{u}_i = -\frac{B_i^2}{12\mu} \overrightarrow{\nabla} P_f \tag{6.32}$$

where

$\vec{u}_i$ =Velocity vector in the fracture plane

B_i =Aperture of i^{th} fracture

$\overrightarrow{\nabla} P_f$=Projection of the pressure gradient in the fracture plane

μ =Fluid viscosity

The pressure gradient in the fracture plane can be expressed as

$$\overrightarrow{\nabla} P_f = \overrightarrow{\nabla} P - \left(\overrightarrow{\nabla} P \cdot \vec{n}_i\right) \vec{n}_i \tag{6.33}$$

where $\overrightarrow{\nabla} P$ is the pressure gradient across the gridblock and $\vec{n}_i$ is the unit normal of i^{th} fracture. The normal unit vector $(\vec{n}_i)$ consists of three componetes; n_x, n_y, and n_z which are the directional cosines of the unit vector.

Expanding Eq. 6.33 yields

$$\overrightarrow{\nabla} P_f = \begin{bmatrix} 1-n_x^2 & -n_x n_y & -n_x n_z \\ -n_y n_x & 1-n_y^2 & -n_y n_z \\ -n_z n_x & -n_z n_y & 1-n_z^2 \end{bmatrix}_i \overrightarrow{\nabla} P \tag{6.34}$$

Substituting Eq. 6.34 into Eq. 6.32 yields

$$\vec{u}_i = -\frac{B_i^2}{12\mu} \begin{bmatrix} 1-n_x^2 & -n_x n_y & -n_x n_z \\ -n_y n_x & 1-n_y^2 & -n_y n_z \\ -n_z n_x & -n_z n_y & 1-n_z^2 \end{bmatrix}_i \overrightarrow{\nabla} P \tag{6.35}$$

Equation 6.35 is the expression for flux through a single fracture. Assuming zero permeability in matrix, the average flux over a domain Ω can be obtained by summing flux over all fractures in the domain and weighting by the fracture void volume:

$$\vec{u}_{average} = \frac{\int\limits_{\Omega} \vec{u}_i dV_i}{V_{bulk}} \tag{6.36}$$

Approximating integral with the summation, Eq. 6.36 can be expressed as

$$\vec{u}_{average} = \frac{\int\limits_{\Omega} \vec{u}_i dV_i}{V_{bulk}} = \frac{\sum\limits_{i=1}^{N_f} \vec{u}_i V_i}{V_{bulk}} \tag{6.37}$$

where, N_f is the total number of fractures in the domain and V_i is the fracture's void volume.

Assuming planar fracture geometry, fracture void volume can be determined by

$$V_i = S_i B_i \tag{6.38}$$

where S_i is the surface area of fracture i.

Substituting Eq. 6.38 into Eq. 6.37 gives

$$\vec{u}_{average} = \frac{\sum_{i=1}^{N_f} \vec{u}_i S_i B_i}{V_{bulk}} \tag{6.39}$$

Substituting Eq. 6.35 into Eq. 6.39 yields

$$\vec{u}_{average} = \frac{\sum_{i=1}^{N_f} \left\{ -\frac{S_i B_i^3}{12\mu} \begin{bmatrix} 1-n_x^2 & -n_x n_y & -n_x n_z \\ -n_y n_x & 1-n_y^2 & -n_y n_z \\ -n_z n_x & -n_z n_y & 1-n_z^2 \end{bmatrix}_i \right\} \vec{\nabla} P}{V_{bulk}} \tag{6.40}$$

On other hand, if the fracture network behaves like a porous medium, then the Darcy's law for the equivalent porous medium can be expressed as

$$\vec{u}_{equiv} = -\frac{\bar{\bar{k}}_{equiv}}{\mu} \vec{\nabla} P \tag{6.41}$$

where $\vec{u}_{equiv}$ is the flux under the pressure gradient $\vec{\nabla} P$ for the equivalent porous medium and $\bar{\bar{k}}_{equiv}$ is the equivalent permeability tensor for the fractured medium.

Comparing Eq. 6.41 and Eq. 6.40 gives an expression for the permeability tensor of the domain. The equivalent permeability tensor can be expressed as

$$\bar{\bar{k}}_{equiv} = \frac{\sum_{i=1}^{N_f} \left\{ -\frac{S_i B_i^3}{12} \begin{bmatrix} 1-n_x^2 & -n_x n_y & -n_x n_z \\ -n_y n_x & 1-n_y^2 & -n_y n_z \\ -n_z n_x & -n_z n_y & 1-n_z^2 \end{bmatrix}_i \right\}}{V_{bulk}} \tag{6.42}$$

Eq. 6.42 shows that the equivalent permeability tensor of a gridblock is a function of the total number of fracture, fracture aperture, fracture surface area, and the gridblock bulk volume.

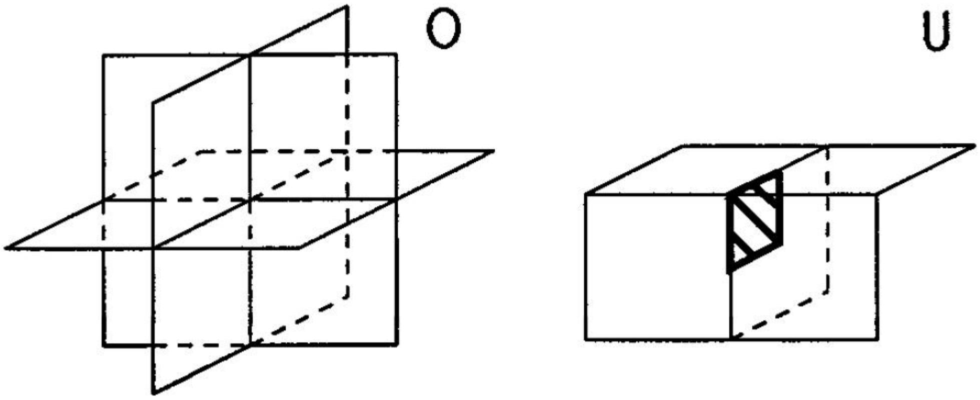


Figure 6.1: Surface with continuity conditions (Aavatsmark *et al.*, 1998)

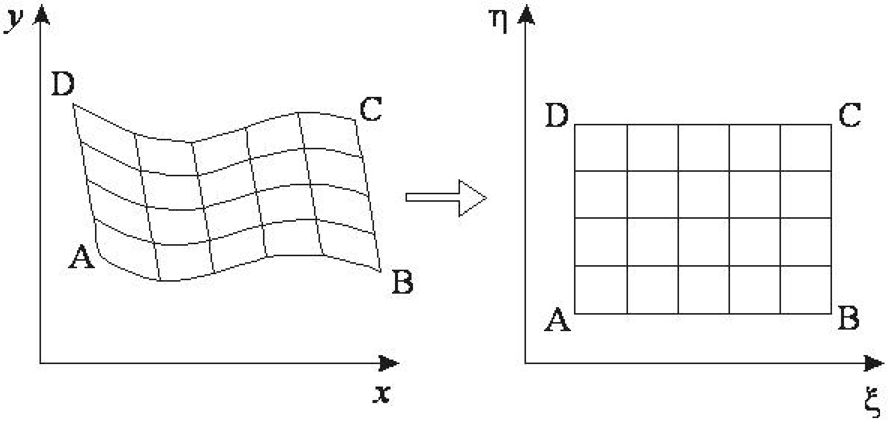


Figure 6.2: Physical and computational domains

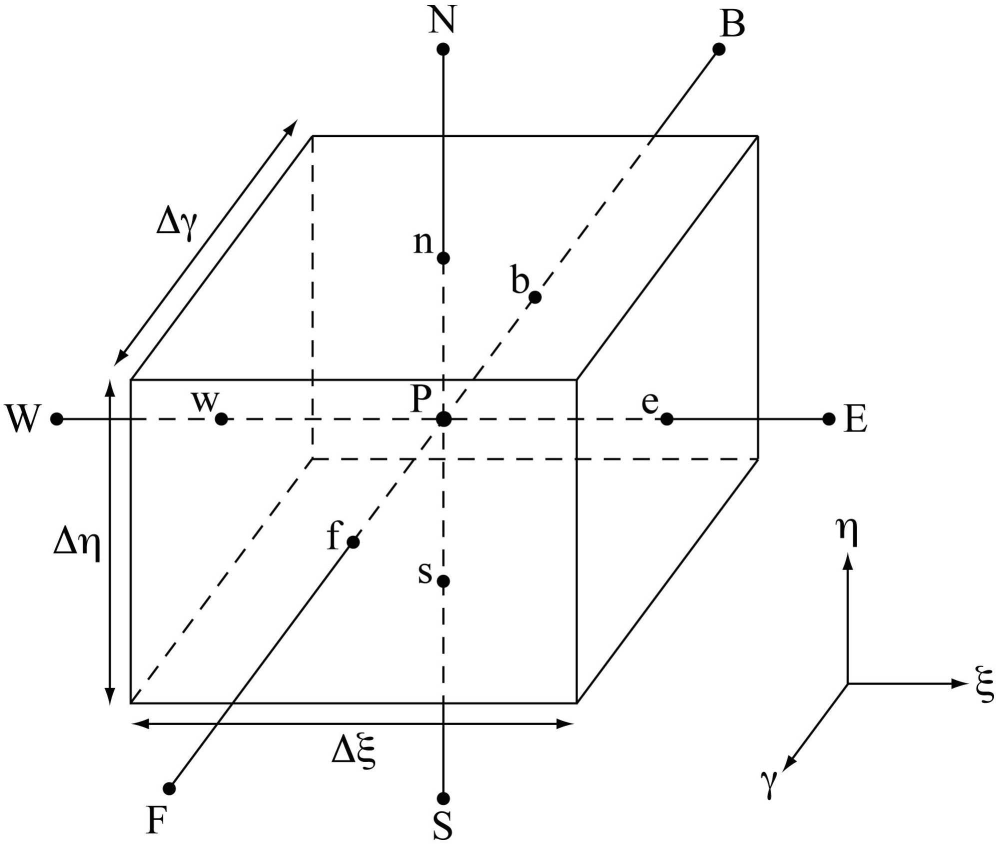


Figure 6.3: Element control volume

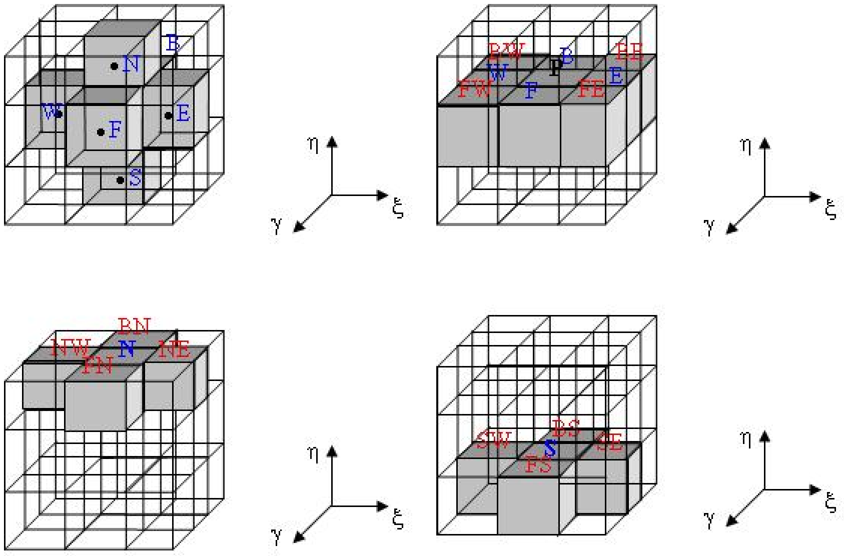


Figure 6.4: 19-point stencil molecule

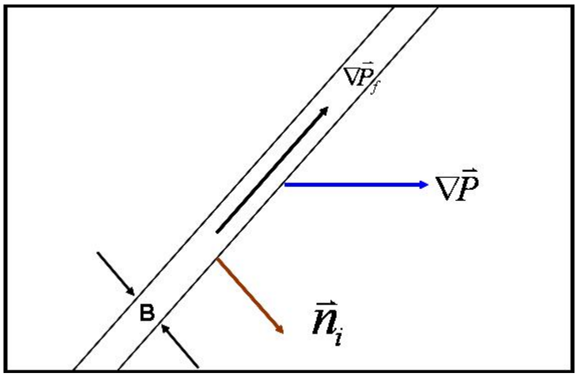


Figure 6.5: Schematic of a fractured used in Oda (1985)

Chapter 7

Verification

Several studies are performed to verify and test the developed dual porosity models and full permeability tensor implementation of GPAS. In the first set of case studies, GPAS results are compared to the results of the UTCHEM simulator for several water and chemical flood cases. Next, the EOS compositional dual porosity model of GPAS is compared with CMG commercial simulator results. Finally, the corner point and full permeability tensor implementations are verified against CMG, ECLIPSE and FracMan commercial simulators.

7.1 Verification of Dual Porosity Models

Validation of the implementation of the dual porosity models are presented in this section. First we verify the implemented chemical dual porosity model against the UTCHEM simulator for several water and chemical flood cases. Although, GPAS already has a module for simulating naturally fractured reservoirs using EOS compositional dual porosity formulation (Naimi-Tajdar *et al.*, 2007), in this work the code was rewritten in order to be compatible with the new development for the full tensor chemical dual porosity module. The EOS compositional dual porosity model is verified against the CMG commercial simulator.

7.1.1 Validation of Chemical Dual Porosity Model

In all of these cases the fully implicit chemical dual porosity option of GPAS is used.

7.1.1.1 1D Waterflood Case

The first validation is based on a 1D waterflood simulation with eight gridblocks in the x direction. The purpose of this simulation is to compare the formulation and implementation of the fully implicit dual chemical model against the UTCHEM simulator. The fracture network consists of 640 ft in x direction, 80 ft in y direction, and 30 ft in z direction. The matrix system consists of matrix blocks of 10x10x10 ft ^{3}. Figure 7.1 shows a fracture system of eight gridblocks where each fracture has a matrix block of four nested subgrids in horizontal direction. An injector is located at the gridblock (1,1,1) operating at a constant rate injection of 498.7 $\frac{STB}{d}$ (2000 $\frac{ft^3}{d}$) and a producer located at gridblock (8,1,1) operating at a constant bottomhole pressure of 3900 $psia$. The initial pressure of the reservoir is 4000 $psia$. Straight line relative permeabilities in fracture media and Corey model relative permeabilities in the matrix media are used (see Fig. 7.2). The physical properties of the fracture and matrix systems and input parameters are shown in Tables 7.1 and 7.2. The GPAS input file for this case is given in Appendix F. The oil and water rates and cumulative oil production vs. time are illustrated in Figs. 7.3 and 7.4, which show an excellent agreement between the results of GPAS and UTCHEM.

One unique feature of these results is the effect of capillary pressure on the recovery and breakthrough time. Water and gas injections are problematic in the fractured reservoir because the injected fluid can channel through fractures without depleting the matrix. The capillary or gravity forces can induce the fluid into the matrix, increasing the oil recovery. Results show an increase of 40% in recovery when capillary pressure is used. Also, the breakthrough time delays significantly.

7.1.1.2 2D Waterflood Case

A modified version of Kazemi *et al.* (1976) quarter-five spot waterflood is investigated to verify the 2D option of the developed dual porosity model of GPAS. In this simulation, water is injected into a quarter-five spot model at the rate of 200 $\frac{STB}{d}$ and liquids are produced at a constant pressure of 3900 $psia$. The simulation domain consists of 600 ft in the x direction, 600 ft in the y direction and 30 ft in the z direction. The fracture system is discretized into 8x8 uniform gridblocks in the x and y directions. A very small initial water saturation of 0.0001 was considered in the fracture media due to the fact that GPAS pressure equation is based on the water phase and we must have water in the system. A schematic of the fractured reservoir is shown in Fig. 7.5. The relative permeability curves for this case is shown in Fig. 7.6. Aldejain (1999) compared the results of UTCHEM and ECLIPSE commercial simulator for a similar case. He showed a very good match between two simulators.

The oil and water rates and cumulative oil production vs. time are shown in Figs. 7.7 and 7.8, respectively. The results show an excellent agreement between GPAS and UTCHEM.

7.1.1.3 3D Waterflood Case

The modified Kazemi *et al.* (1976)'s quarter-five spot waterflood model used in the previous section was extended to 3D to verify the 3D option of the developed dual porosity model. The thickness of the reservoir was divided into six layers of 5 ft thickness for the fracture network. Also, the size of the matrix block was changed to 10x10x5 ft^3. All other physical properties and input parameter were kept the same.

The results of GPAS dual porosity model are compared with the results of the UTCHEM simulator. The oil and water rates and cumulative oil production vs. time are demonstrated in Figures 7.9 and 7.10, respectively. The results show an excellent agreement between GPAS and UTCHEM.

7.1.1.4 Tracer Case Study

To verify the correctness of the implemented tracer model, a single-layer quarter-five-spot reservoir is used. The input parameters are similar to the ones used in the 2D waterflood case. The only exceptions are that the residual water saturation in fracture is increased from 0 to 10% and the matrix permeability is increased from 1 md to 5 md. A conservative tracer slug of 0.052 PV (200 days) is injected at the rate of 100 $\frac{STB}{d}$ followed by water only at the same rate for 1800 days.

The main driving forces in fractured reservoirs are the counter-current imbibition by capillary pressure and the co-current imbibition by gravity force. In this particular example the gravity effect is turned off so the only driving mechanism to expel oil from the matrix is the capillary force. The effluent responses for GPAS and UTCHEM are shown in Fig. 7.11. The results of GPAS and the UTCHEM simulator are in great agreement. Since the conservative tracer remains only in water, it flows into the matrix blocks by capillary force and traps in the matrix blocks. Hence, the conservative tracer test can be used to investigate if counter-current or co-current imbibition is taking place in the matrix.

7.1.1.5 Surfactant Case Study

The effect of surfactant is to lower interfacial tension (IFT) between oil and water. Lowering IFT diminishes the capillary forces in the system which is the main driving force in fracture reservoirs. Aldejain (1999) has addressed several issues with surfactant flooding in fractured reservoirs. Based on his study, injecting the surfactant at the beginning of the waterflood reduces the ultimate oil recovery. The best way to perform surfactant flooding in fractured reservoirs is to allow for the recovery by capillary and gravity forces and then use surfactant.

This test is designed to verify the GPAS surfactant flood option. The same case used in the tracer flood study was used for surfactant flood. To have the shorter waterflood period, the matrix porosity was reduced from 19% to 9% in this case. The water was injected at 577.7 $\frac{STB}{d}$ for the first 1600 days followed by a chemical solution with a surfactant concentration of 0.01 volume fraction for 2400 days. The properties of the surfactant used in this study and relative permeability parameters are listed in Table 7.3. The oil recovery vs. time for GPAS and UTCHEM are shown in Fig. 7.12. The results show a good agreement between GPAS and UTCHEM simulators.

7.1.2 Validation of EOS Compositional Dual Porosity Model

7.1.2.1 Gas Injection

The SPE fifth comparative solution problem (Killough and Kossack, 1987) was converted to a dual porosity model to verify the EOS compositional dual porosity model of GPAS. The simulation domain consists of 560 ft in length, 560 ft in width, and 100 ft in thickness. A 7x7x3 gridblock configuration is used for the fracture system with the matrix block size of 10x10x10 ft ^{3}. The three layers have 20, 30, and 50 ft thickness from top to bottom, respectively. Each matrix block is discretized in 2x1 subgrids. The reservoir fluid consists of six hydrocarbon components. The initial composition and properties of components are given in Tables 7.4 and 7.5. The input parameters for this case are presented in Table 7.6. An injector is located at gridblocks (1,1,1) through (1,1,3) operating at a constant rate injection of 1.0 $\frac{Mscf}{d}$ and a producer located at gridblocks (7,7,1) through (7,7,3)operating at a constant bottomhole pressure of 1300 $psia$.

The CMG's general EOS-based compositional reservoir simulator (GEM) is used to verify the results of GPAS. Oil recovery and oil and gas production rates vs. time are shown in Figs. 7.13 through 7.15. The results show a good agreement between GPAS and the GEM simulator.

7.2 Verification of Corner Point Implementation

As discussed earlier (see Section 6.3 on page 115), the full permeability tenor has been implemented based on a corner point mesh formulation. In this case study, we carried out a comparison between GPAS and the CMG-GEM simulator using an irregular reservoir with 45x24x4 gridblocks in conjunction with the corner point implementation. There are four injection and four production wells in this case (see Fig. 7.16). Water is injected at a constant rate of 6000 $\frac{STB}{d}$ (1500 $\frac{STB}{d}$ per each well) for 25,000 days. The producers are operating at a constant bottomhole flowing pressure of 1300 $psia$. The reservoir has an isotropic constant permeability of 100 md and a constant porosity of 0.35. The fluid and reservoir properties are given in Table 7.7. The relative permeability curves for this case are shown in Fig. 7.17.

The field oil and water production rates and oil recovery vs. time for two simulators are shown in Figs. 7.18 and 7.19. There is a good agreement between two simulators. However, the CMG-GEM results show a fluctuation between 7500 to 9500 days. Regardless of reviewing the input files and simulation output, I could not find any reason for this fluctuation. Figures 7.20 and 7.21 show the water saturation maps at the end of simulation time (25,000 days) for GPAS and the CMG-GEM, respectively. The saturation maps are almost identical between the two simulators. The difference in colors between two figures is because they are generated by different packages.

7.3 Verification of Full Permeability Tensor Implementation

The full permeability tensor was implemented for both chemical and EOS compositional models in GPAS. In this section, we verify the validation of the implemented full permeability tensor. First, we test the full permeability tensor option of GPAS to ensure that the full permeability model with zero off diagonal terms reproduces the same results as diagonal permeability model. Next, the full permeability tensor of GPAS is verified against the ECLIPSE commercial simulator. Finally, the model is verified for 2D and 3D problems on cylindrical and spherical reservoirs.

7.3.1 Full Tensor Option with Diagonal Permeability

Diagonal permeability tensors are used when the principle directions of permeabilities are aligned with the simulation direction. For comparison, test cases were run using both diagonal dual porosity and full tensor dual porosity options. The 3D waterflood (Case 7.1.1.3), 2D surfactant flood (Case 7.1.1.5), and the 3D gas injection (Case 7.1.2.1) cases were chosen for comparison. The full permeability runs were made with zero off-diagonal terms. Comparison between two models are shown in Figs. 7.22 through 7.24. The results show that both models give identical results.

7.3.2 Validation Against ECLIPSE Simulator

ECLIPSE commercial simulator supports a full tensor description of absolute permeability. ECLIPSE uses a multipoint flux approximation to construct the flow equations. The full permeability option in ECLIPSE is not compatible with its dual porosity model, therefore, the ECLIPSE runs are single porosity runs. In order to make the same run with GPAS, the permeability and porosity of the matrix system are chosen to be very small in order to ignore the effect of the matrix system and be able to use the dual porosity model to simulate a single porosity model.

Comparisons were made for a 2D and a 3D cases. E300, compositional simulator of the ECLIPSE software suite, is used for the comparison. A 600x600x30 ft ^{2} homogeneous reservoir was chosen for the test. In the 2D case, the reservoir is discretized into a 20x20x1 gridblock system. An injector is located at the gridblock (1,1,1) injecting at a constant rate of 200 $\frac{STB}{d}$, and a producer is located at the gridblock (20,20,1) operating at a constant bottomhole pressure of 1500 $psia$. The initial reservoir pressure is 2000 $psia$. The description of the reservoir is given in Table 7.8. A 2D full permeability tensor of $\bar{\bar{k}} = \begin{bmatrix} 500 & 200 \\ 200 & 300 \end{bmatrix}$ is used in this simulation run. Figure 7.25 and 7.26 show the oil recovery and the oil and water production rates for both GPAS and E300. The results show that the GPAS full permeability tensor implementation gives the same results as the ECLIPSE full permeability tensor option.

To test the full permeability tensor implementation in 3D, the reservoir used in the 2D case is discretized in a 20x20x5 gridblock system. All physical and fluid properties are the same as those used in the 2D case except the permeability tensor in 3D is given by

$$k = \begin{bmatrix} 443.13 & 493.5 & 13.44 \\ 493.5 & 771.8 & -21.9 \\ 13.44 & -21.9 & 1090.226 \end{bmatrix}$$

The oil recovery and the oil and water production rates vs. time are shown in Figs. 7.27 and 7.28 for this case. The results confirm the validation of the full permeability tensor implementation in GPAS.

7.3.3 Cylindrical Reservoir

Whenever the principle directions of permeability are not aligned with the coordinate system in an anisotropic system, a full permeability tensor results. Ignoring the off-diagonal terms of the permeability tensor in such cases, yields significant errors. Theoretically, using a full permeability tensor in cases that the coordinate system is rotated should give the same results as the original coordinate system with diagonal permeability tensor.

A cylindrical reservoir is designed to verify the full permeability tensor option of GPAS in 2D. A rectangular domain of 1000x1000 ft ^{2} with the thickness of 100 ft is selected for this test. An 81x81x1 gridding scheme is used. To have a cylindrical system, all gridblocks outside of the circle are set as inactive grids. Figure 7.29 shows a schematic of the reservoir. The total number of gridblocks is 6561, which 1392 of them are inactive grids. Two wells are placed symmetrically on opposite sides of a producer which is located at the center of reservoir. The initial reservoir pressure is 4000 *psia* and the producer operates at a constant liquid production rate of 400 $\frac{STB}{d}$. Both injectors have the same injection rate of 200 $\frac{STB}{D}$. The reservoir is homogeneous and anisotropic with the permeability contrast of 1:10, *i.e.* k_{xx} = 50 md, k_{yy} = 500 md, and k_{zz}= 500 md. Input parameters for this model are given in Table 7.9. A waterflood is performed for this system. In the next scenario, the coordinate system of simulation is rotated 45 degrees (Fig. 7.30). As a result, a full permeability tensor yields for the reservoir under rotation. The new permeability tensor after 45 degrees rotation is (see Appendix D for a complete detail of permeability tensor transformation).

$$\bar{\bar{k}} = \begin{bmatrix} 275 & -225 & 0 \\ -225 & 275 & 0 \\ 0 & 0 & 500 \end{bmatrix}$$

Note that even though the physical locations of the wells are the same, the locations of the wells in the new coordinate system are different than their location in the original coordinate system. A new simulation run is performed for the new case with the full permeability tensor and the results are compared with those of the original coordinate system. In Fig. 7.31, curves labeled *Before rotation* are the oil and water production rates for the original system, and curves labeled *Full tensor after rotation* are the oil and water production rates for the rotated system. The results show an excellent agreement between the two runs.

A similar simulation is performed while ignoring the off-diagonal terms of the permeability tensor in order to show the case if we do not use full permeability tensor while the principal direction of permeability is not aligned with the coordinate system. The water and oil production rates for this case are compared with the results of the case with full permeability tensor in Fig. 7.32. Results show that great care must be taken when dealing with permeability anisotropy.

To ensure the validation of the full permeability tensor implementation in the chemical model, similar simulation runs are performed for a chemical flood process in a dual porosity system. The same cylindrical domain is converted to a dual porosity system with the fracture porosity of 0.01 and the matrix porosity of 0.19. The diagonal permeability of 5 md are chosen for the matrix system. For the fracture system, the same permeability sets used in the previous waterflood runs are used for the original and rotated coordinate systems. A waterflood is performed for the first 3000 days to allow for the maximum recovery by capillary force, followed by a chemical solution with a surfactant concentration of 0.05 volume fraction for the next 2000 days. The water and oil production rates vs. time for the original and rotated coordinate systems are shown in Fig. 7.33. The results show a satisfactory agreement. Also, to show the effect of off-diagonal terms in the simulation results, a similar run is performed with zero off-diagonal permeability term while the coordinate system is rotated. Results for this case are plotted along the results of the case with full permeability tensor after rotation in Fig. 7.34. As shown in Fig. 7.34, improper handling of permeability anisotropy produces significant errors.

7.3.4 Effect of Coordinate Misalignment

When the principle directions of a permeability tensor are not aligned with the simulation coordinate system, a full permeability tensor results. Ignoring off-diagonal terms of a permeability tensor will introduce an error where a full tensor exists. The misalignment angle can be considered by the angle between the anisotropy and the simulation coordinate axes. To investigate the effect of the misalignment, simulations were performed considering two scenarios.

The same cylindrical reservoir described in Section 7.3.3 is used for this study. The original permeability tensor (before rotation) had an anisotropy ratio of 1:10 (the permeability component in y direction was 10 times of the permeability component in x direction). Two scenarios were considered in this study. In the first scenario, the original diagonal permeability tensor was used while the simulation coordinate system was rotated θ degree. In the second scenario, a full permeability tensor was calculated after rotation but the off-diagonal terms of permeability tensor were intentionally set to zero to reveal their effects on the simulation results. In both scenarios, waterfloods were performed and the water breakthrough times were recorded.

Table 7.10 lists the water breakthrough time for different rotation angles for the first scenario. In this scenario, the original diagonal permeability tensor was used. The normalized breakthrough time in Table 7.10 is the ratio of the breakthrough time of the original system divided by the breakthrough time when the system is rotated θ degree. The results are also plotted in Fig. 7.35. The results show a slight deviation of 10% until the rotation angle of 20°. For a rotation angle of 25° and higher, the discrepancy increases to 16% and higher.

The breakthrough times for the second scenario are listed in Table 7.11. In this scenario, a full permeability tensor was calculated after the rotation and the off-diagonal terms of the resulting permeability tensor were set to zero. The results are also plotted in Fig. 7.36. The results show that for the anisotropy contrast of 1:10, until 15° the difference is only about 7%, and the discrepancy increases to 40% and higher for misalignment angles of 20° or more.

7.3.5 Spherical Reservoir

Analogues to the cylindrical model used for the verification for 2D problem, a spherical reservoir model was designed to verify the implementation of full permeability tensor in 3D. The reservoir dimension is 400 ft in length, 400 ft in width, and 400 ft in thickness discretized in a 45x45x45 gridblocks system. The spherical reservoir is created using a cube whose size has the same length as the diameter of the sphere. All gridblocks outside of the sphere are set as inactive gridblocks. There are 47833 active gridblocks in this system. Two injection wells are placed symmetrically on the sides of the producer which is located at the center of the sphere. To avoid well deviation after rotation, each well is perforated only in one gridblock. Each injection well has a constant injection rate of 100 $\frac{STB}{d}$ and the producer operates at a constant bottomhole flowing pressure of 3900 $psia$. The schematic of the reservoir is shown in Fig. 7.37. The reservoir under the original system has the following diagonal permeability tensor

$$\bar{\bar{k}} = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 300 & 0 \\ 0 & 0 & 35 \end{bmatrix}$$

Three runs are performed in this case. In the first run, the principle direction of the permeability field is aligned with the coordinate system. The second run is performed after the coordinate system is rotated 45° in x-y and then 15° in y-z planes sequentially. A full permeability tensor is yield with the sequential rotations of the coordinate system in at least two planes (see Appendix D). The new permeability tensor under the new coordinate system becomes

$$\bar{\bar{k}}' = \begin{bmatrix} 200 & -96.59 & -25.88 \\ -96.59 & 188.95 & 41.25 \\ -25.88 & 41.25 & 46.05 \end{bmatrix}$$

In the third run, the off-diagonal terms of the $\bar{\bar{k}}'$ are set to zero; this gives a permeability tensor

$$\bar{\bar{k}}'' = \begin{bmatrix} 200 & 0 & 0 \\ 0 & 188.95 & 0 \\ 0 & 0 & 46.05 \end{bmatrix}$$

Note that, in all these cases the gravity term is turned off. The gravity always acts in the vertical direction and in GPAS there is no such option to rotate the direction of the gravity vector in an arbitrary direction. The results for the three cases are plotted in Fig. 7.38. The results show a good agreement between full permeability tensor after rotation and the results of diagonal permeability tensor before rotation. Also, the results show a significant difference if the off-diagonal terms of the permeability tensor are ignored in the case that the principle direction of permeability is not aligned with the simulation coordinate system. A relatively small difference between the water breakthrough time in this case is due to the fact that more gridblocks are needed to minimize the difference.

7.3.6 Validation with a Discrete Fracture Network Simulator

The purpose of this study is to use GPAS full permeability tensor option to simulate flow in a fracture reservoir and compare the GPAS results with results of a discrete fracture network simulator. A comparison between results from GPAS and results from FracMan simulator (Golder Associate, 2007) are made. The FracMan software is a flow simulator for fractured reservoir based on discrete fracture. FracMan includes tools for discrete feature data analysis, geologic modeling, spatial analysis, visualization, flow and transport, and geomechanics. The FracMan simulator is used for analysis of flow and solute transport through three-dimensional rock masses with discrete fracture network. The FracMan suite consists of several packages; the FracMan/FracWorks to generate and analyze fracture models, the MeshMaker to generate finite element meshes based on fracture geometries generated by FracMan/FracWorks and boundary conditions, and a Matrix/Fracture Interactive Code (MAFIC) to simulate the transient flow and solute transport through rock masses. The workflow starts by generating a fracture network using FracMan/FracWorks and then generating a finite element mesh using the MeshMaker. MeshMaker incorporates the boundary conditions and other flow characteristics into the fracture networks and generates an input file for the flow simulator. Finally, the flow simulator takes the input file and simulates flow and solute transport in the fracture and rock matrix.

In this case study, GPAS was used to simulate waterflooding of a naturally fractured reservoir. The results of GPAS are compared with the results of the MAFIC flow simulator. A real fracture system was generated using the FracMan/FracWorks package, see Fig. 7.39. The reservoir dimension is 600 ft in length, 600 ft in width, and 30 ft in thickness. There are 60 fractures in this system with the average fracture aperture of 3.28×10^{-4} ft. The total void volume in this system is about 99 barrels. One injector and one producer are placed at two corners of the reservoir (see Fig. 7.39). In the FracMan model, well radius are larger than normal to have more intersecting fractures. FracMan can only simulate a single phase (water phase) system. FracMan uses a particle tracing algorithm to simulate tracer in fractured systems.

A tracer flood simulation was performed using GPAS full permeability tensor option and the MAFIC package of the FracMan suite. The reservoir is saturated 100% with water and water is injected at a constant rate at the injector. A tracer slug of 0.02 pore volume was injected into the reservoir. For the GPAS run, a 20x20x1 gridblock scheme was used. The equivalent gridblock permeability tensor was calculated using Oda's method (Oda, 1985) (see Section 6.5 on page 126 for more details). Figure 7.40 shows the equivalent gridblock permeability for the GPAS model. The tracer concentration vs. time for both GPAS and FracMan are shown in Fig. 7.41. Results are in good agreement and show that the full permeability tensor is a key parameter to capture the heterogeneity and orientation of fractures.

Table 7.1: Fracture input parameters used in 1D waterflood for Case 7.1.1.1

Number of gridblocks	8x1x1
Size of gridblocks	75x75x30 ft^3
Porosity	0.01
Permeability	500 md
Initial water saturation	0.0
Water viscosity	0.5 cp
Oil viscosity	2.0 cp
Initial reservoir pressure	4000 psia
Injection rate	200.0 $\frac{STB}{d}$
Production pressure	3900 psia

Table 7.2: Matrix input parameters used in 1D waterflood for Case 7.1.1.1

Number of matrix subgrids	4x4
Size of matrix subgrids	10x10x10 ft^3
Porosity	0.19
Permeability	1 md
Initial water saturation	0.25
Water viscosity	0.5 cp
Oil viscosity	2.0 cp
Residual oil saturation	0.3
Endpoint relative permeability of oil	0.92
Corey's exponent for water	1.8
Residual water saturation	0.25
End point relative permeability of water	0.2
Corey's exponent for water	1.18
Initial reservoir pressure	4000 psia

Table 7.3: Surfactant and relative permeability parameters for Case 7.1.1.5

Description	Fracture	Matrix
Heights of bimodal curve	0.07, 0.04 volume fraction	
Salinity limits	0.177, 0.25 $\frac{meq}{ml}$	
CMC	0.0001	
Adsorption parameters	1.5, 0.5, 1000	
IFT correction parameters	9.0, 2.0	
Low trapping number relative permeability	Residual saturations: 0.0 0.0 0.0	0.25 0.30 0.25
	Exponents : 1.46 2.15 1.0	1.18 1.80 1.18
	Endpoints : 1.00 1.00 1.0	0.20 0.92 0.20
High trapping number relative permeability	Residual saturations:	0.00 0.00 0.00
	Exponents	1.18 1.80 1.18
	Endpoints	1.00 1.00 1.00

Table 7.4: Initial composition and properties of components used in Case 7.1.2.1

Property	C_1	C_3	C_6	C_{10}	C_{15}	C_{20}
Initial composition	0.5	0.03	0.07	0.2	0.15	0.05
Injected gas composition	0.77	0.20	0.01	0.01	0.005	0.005
T_c ($^\circ$ R)	343.0	665.7	913.4	1111.8	1270.0	1380.0
P_c (psia)	66.7.8	616.3	439.9	304.0	200.0	162.0
$V_c\ \frac{ft^3}{lbmol}$	1.599	3.211	5.923	10.087	16.696	21.484
Z_c	0.29	0.277	0.264	0.257	0.245	0.235
Molecular Weight	16.0	44.1	86.2	142.3	206.0	282.0
Acentric Factor	0.013	0.152	0.301	0.488	0.650	0.850
Parachors	71.0	151.0	271.0	431.0	631.0	831.0

Table 7.5: Binary interaction coefficients for Case 7.1.2.1

BIC	C_1	C_3	C_6	C_{10}	C_{15}	C_{20}
C_1	0.0	0.0	0.0	0.0	0.05	0.05
C_3	0.0	0.0	0.0	0.0	0.005	0.005
C_6	0.0	0.0	0.0	0.0	0.0	0.0
C_{10}	0.0	0.0	0.0	0.0	0.0	0.0
C_{15}	0.05	0.005	0.0	0.0	0.0	0.0
C_{20}	0.05	0.005	0.0	0.0	0.0	0.0

Table 7.6: Input parameters for gas injection (Case 7.1.2.1)

Description	Fracture	Matrix
Number of gridblocks	7x7x3	2x1
Porosity	0.01	0.35
Permeability	$K_{xx} = 50$ md $K_{yy} = 50$ md $K_{zz} = 5$ md	$K_{xx} = 1$ md $K_{yy} = 1$ md $K_{zz} = 1$ md
Initial water saturation	0.01	0.17
Water viscosity	1.0 cp	1.0 cp
Residual oil saturation	0.0	0.1
Endpoint relative permeability for oil	1.0	0.9
Corey's exponent for oil	1.0	2.0
Residual gas saturation	0.0	0.0
Endpoint relative permeability for gas	1.0	0.9
Corey's exponent for oil	1.0	2.0
Residual water saturation	0.0	0.3
Endpoint relative permeability for water	1.0	0.4
Corey's exponent for water	1.0	3.0
Initial reservoir pressure	1500 psia	
Gas injection rate	$1.0\ \frac{Mscf}{D}$	
Production well pressure	1300 psia	

Table 7.7: Input parameters for corner point case, Case 7.2

Reservoir area	$3.73462\times10^5 ft^2$
Reservoir thickness	50 ft
Number of gridblocks	45x24x4
Porosity	0.35
Permeability	100 md
Initial compositions	$C_{10} = 0.7$ $C_{15} = 0.2$ $C_{20} = 0.1$
Reservoir temperature	160 °F
Initial water saturation	0.17
Water viscosity	1.0 cp
Initial reservoir pressure	1500 psia
Water injection rate	6000 $\frac{STB}{d}$
Production well pressure	1300 psia

Table 7.8: Input parameters for the comparison between GPAS and ECLIPSE for the full permeability tensor option (Case 7.3.2)

Number of gridblocks	20x20x1
Size of gridblocks	30x30x30 ft^3
Porosity	0.2
Permeability	k_{xx} = 500 md k_{yy} = 300 md k_{xy} = 200 md k_{zz} = 500 md
Initial water saturation	0.20
Water viscosity	1.0 cp
Initial reservoir pressure	2000 psia
Injection rate	200 $\frac{STB}{d}$
Production pressure	1500 psia
Total simulation time	2000 days

Table 7.9: Input parameters for the 2D cylindrical system (Case 7.3.3)

Description	Parameter
Number of gridblocks	81x81x1
Size of gridblocks	12.35x12.35x100 ft^3
Porosity	0.2
Permeability	$k_{xx} = 50$ md $k_{yy} = 500$ md $k_{zz} = 500$ md
Initial water saturation	0.25
Water viscosity	1.0 cp
Initial reservoir pressure	4000 $psia$
Injection rate	400 $\frac{STB}{d}$
Production rate	400 $\frac{STB}{d}$
Total simulation time	3000 days

Table 7.10: Water breakthrough time for different rotation angles using original permeability tensor is used (Case 7.3.4)

Rotation degree	Breakthrough time (sec)	Normalized breakthrough time
0	1305	1.000
5	1298	1.005
10	1301	1.003
15	1245	1.048
20	1184	1.102
25	1123	1.162
30	1071	1.218
35	1015	1.286
40	960	1.359
45	843	1.548

Table 7.11: Water breakthrough time for different rotation angles ignoring off-diagonal terms of permeability tensor (Case 7.3.4)

Rotation degree	Breakthrough time (sec)	Normalized breakthrough time
0	1305	1.000
5	1280	1.020
10	1223	1.067
15	1218	1.071
20	939	1.390
25	808	1.615
30	716	1.823
35	658	1.983
40	630	2.071
45	605	2.157

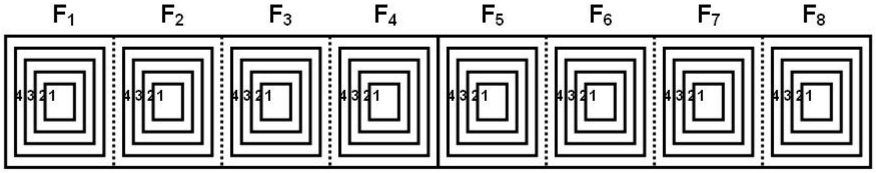


Figure 7.1: Schematic of the discretized fracture media for Case 7.1.1.1

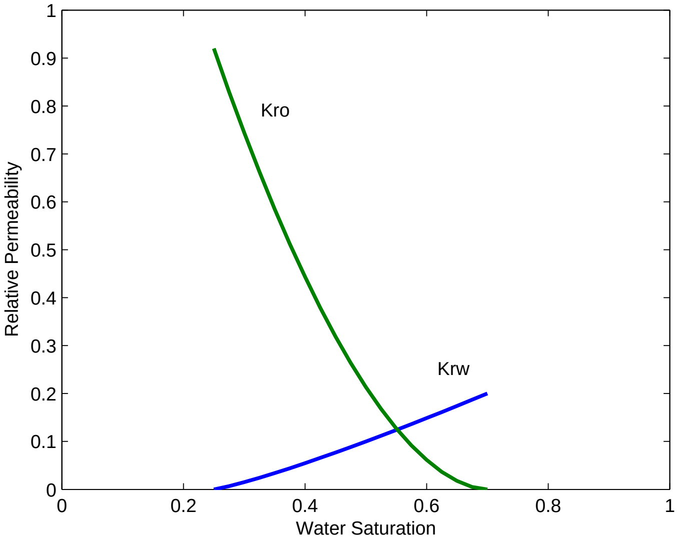


Figure 7.2: Matrix relative permeability curves for Case 7.1.1.1

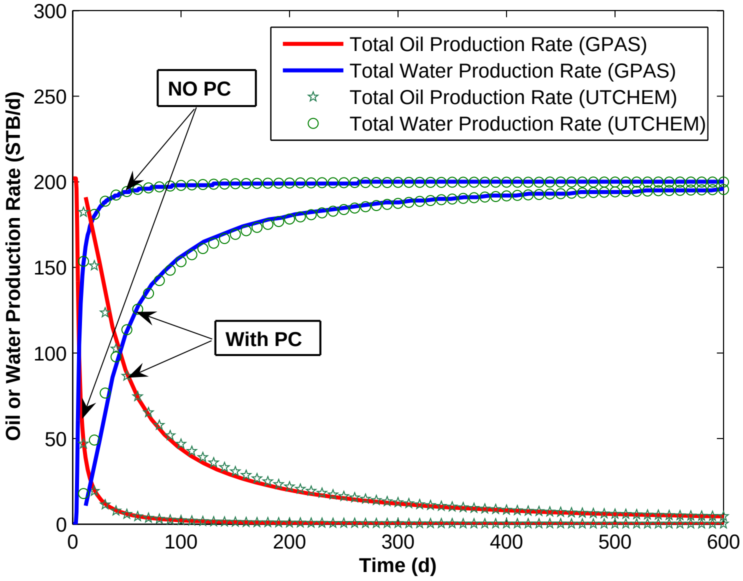


Figure 7.3: Water and oil production rates for Case 7.1.1.1

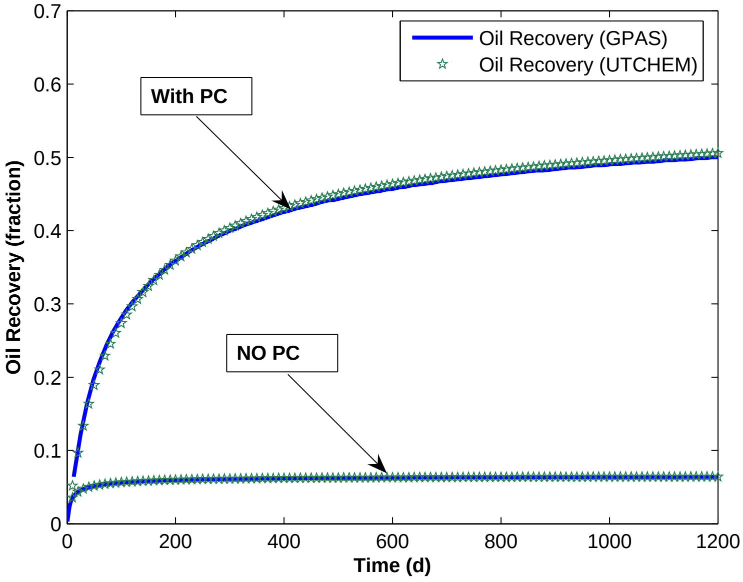


Figure 7.4: Oil recovery vs. time for Case 7.1.1.1

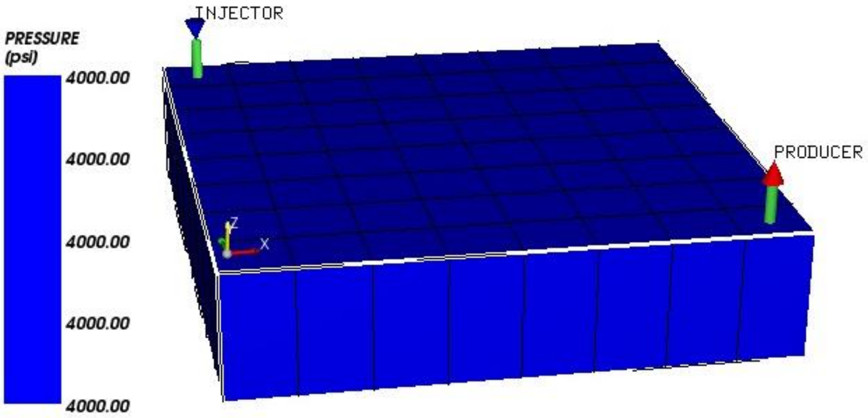


Figure 7.5: Schematic of the fractured reservoir used in the quarter-five-spot case, Case 7.1.1.2

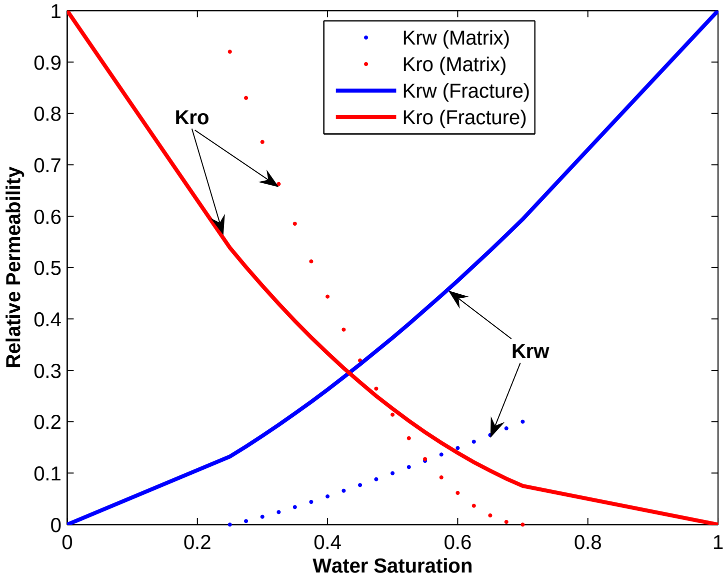


Figure 7.6: Fracture and matrix relative permeabilities used in Case 7.1.1.2

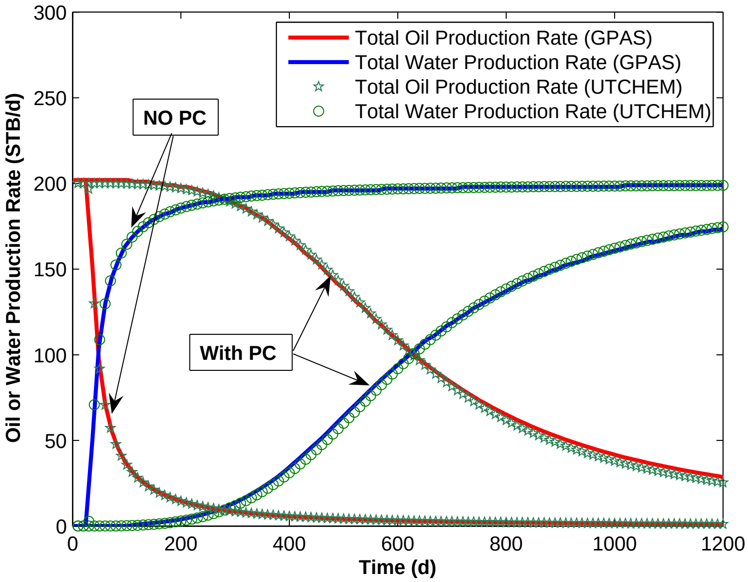


Figure 7.7: Water and oil production rates for Case 7.1.1.2

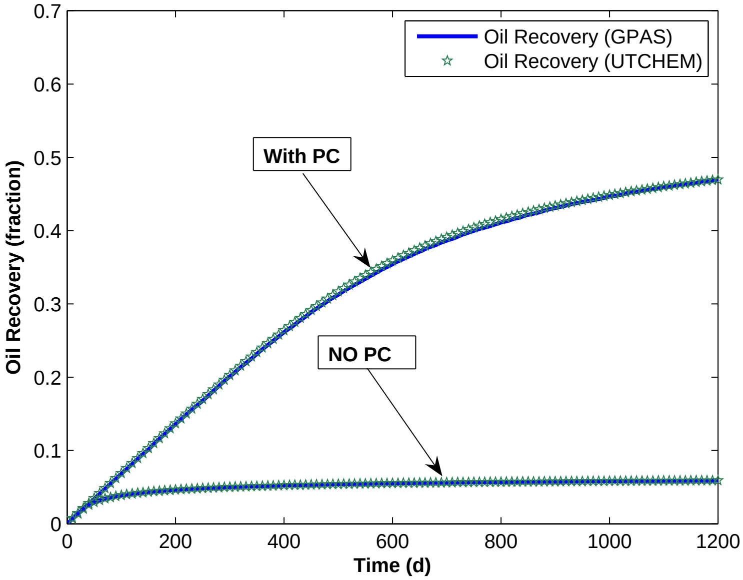


Figure 7.8: Oil recovery vs. time for Case 7.1.1.2

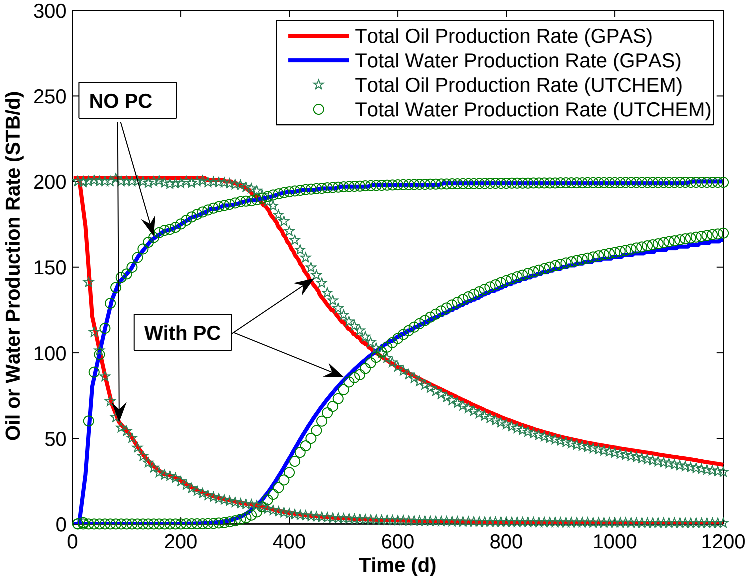


Figure 7.9: Water and oil production rates for Case 7.1.1.3

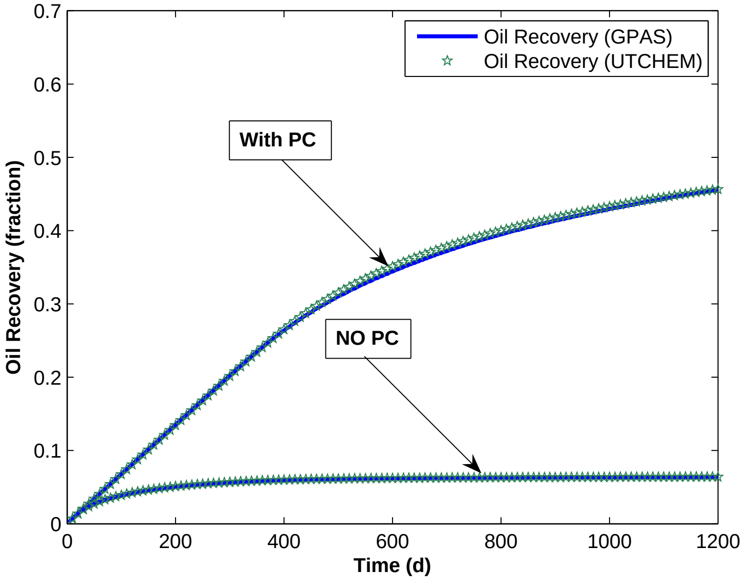


Figure 7.10: Oil recovery vs. time for Case 7.1.1.3

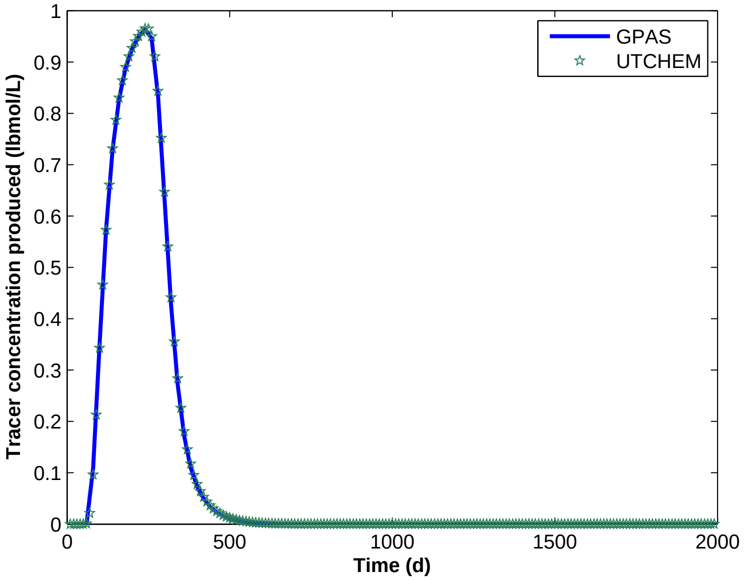


Figure 7.11: Effluent concentration for 2D tracer case

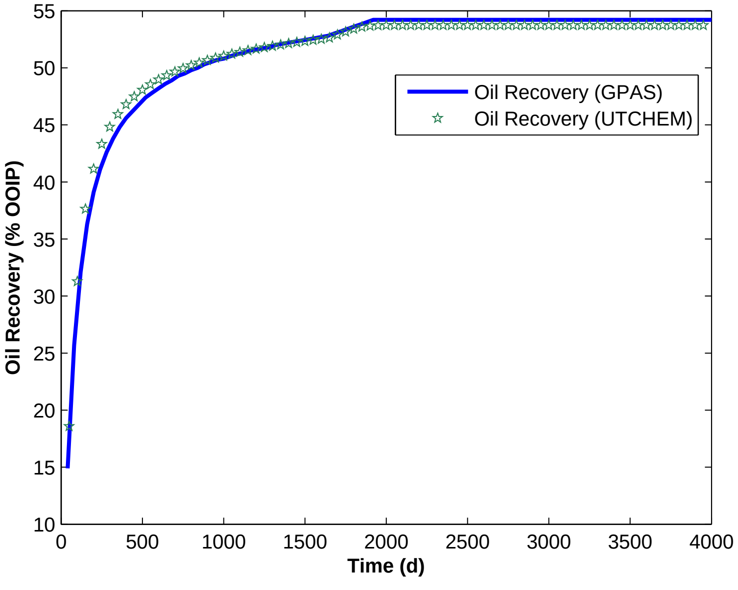


Figure 7.12: Surfactant flood oil recovery for GPAS and UTCHEM

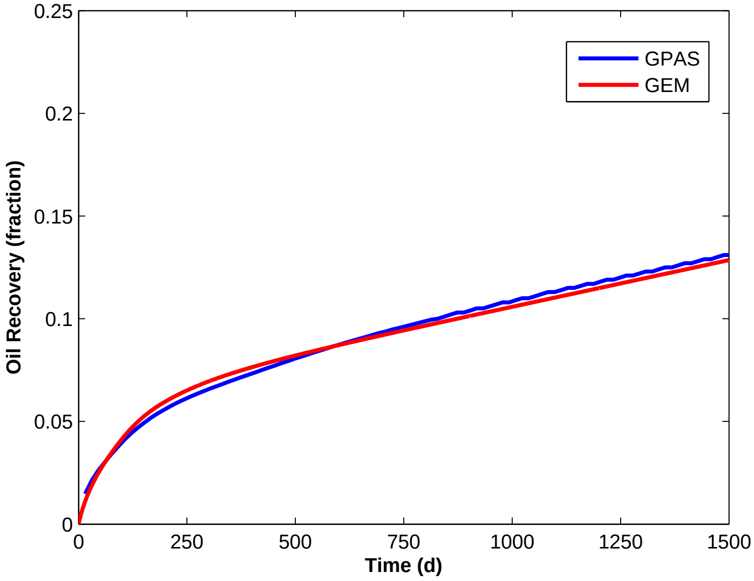


Figure 7.13: Oil recovery vs. time for Case 7.1.2.1

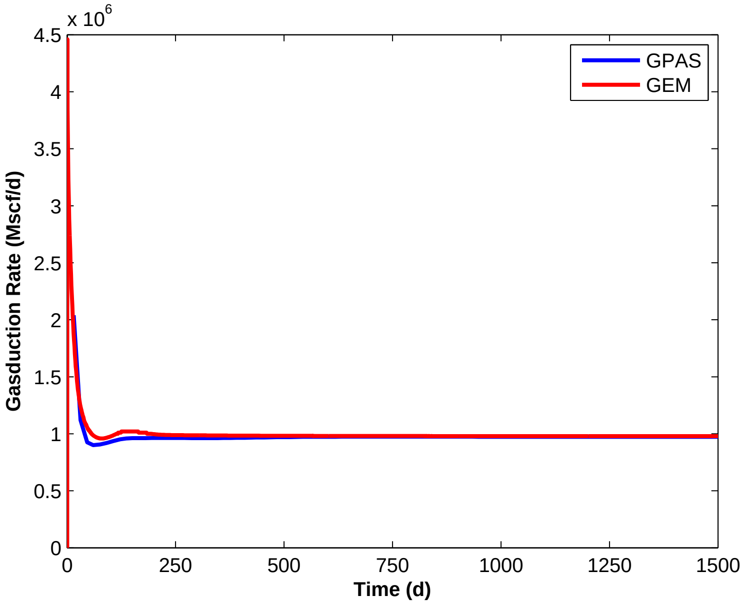


Figure 7.14: Gas rate vs. time for Case 7.1.2.1

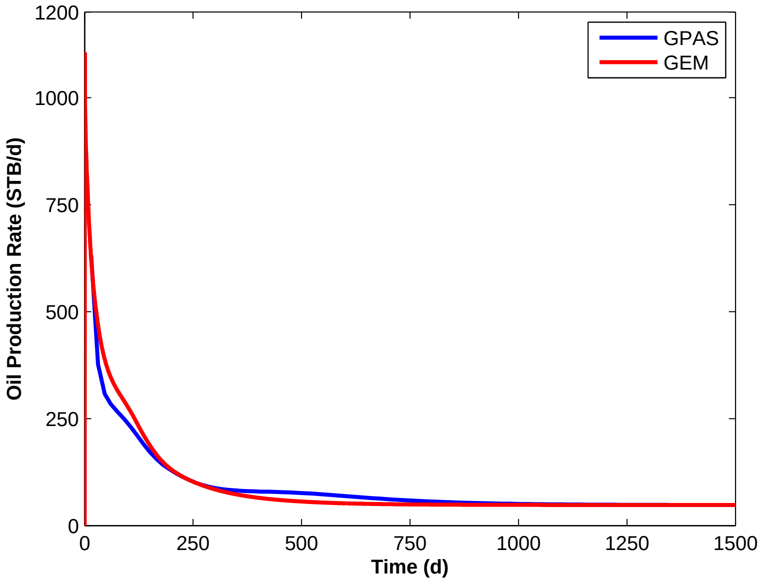


Figure 7.15: Oil rate vs. time for Case 7.1.2.1

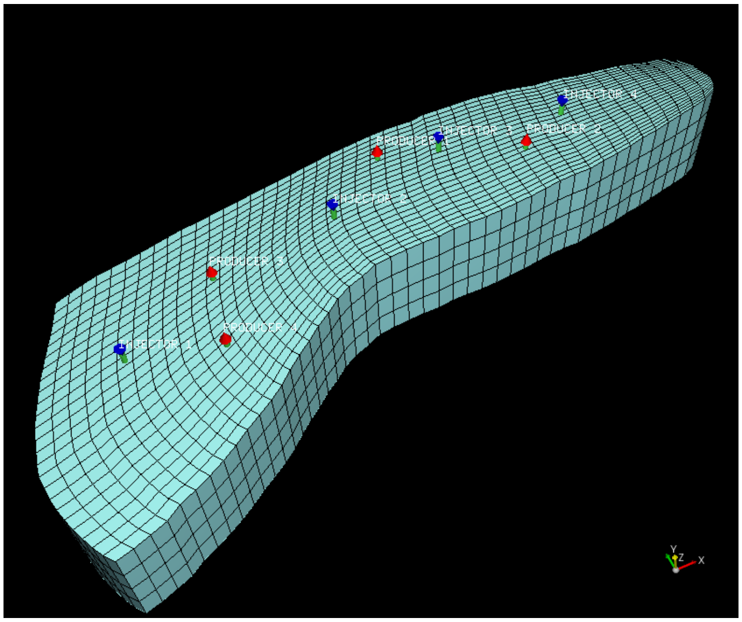


Figure 7.16: Schematic of reservoir for corner point case, Case 7.2

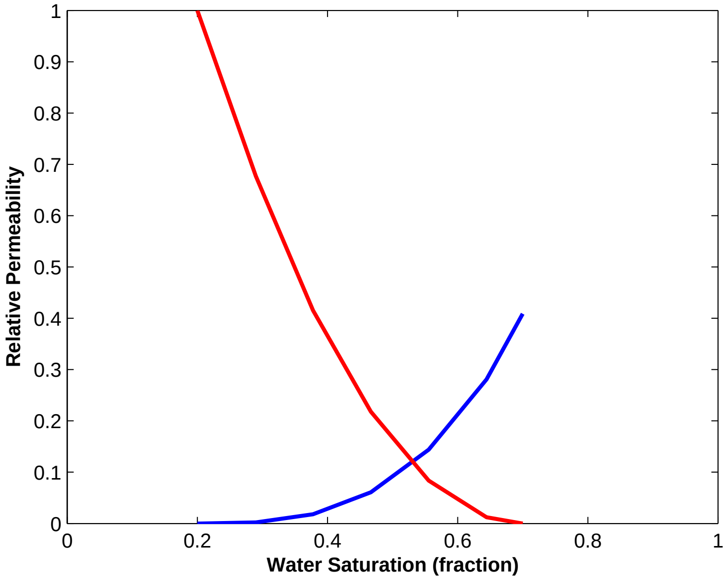


Figure 7.17: Relative permeability curves for corner point case, Case 7.2

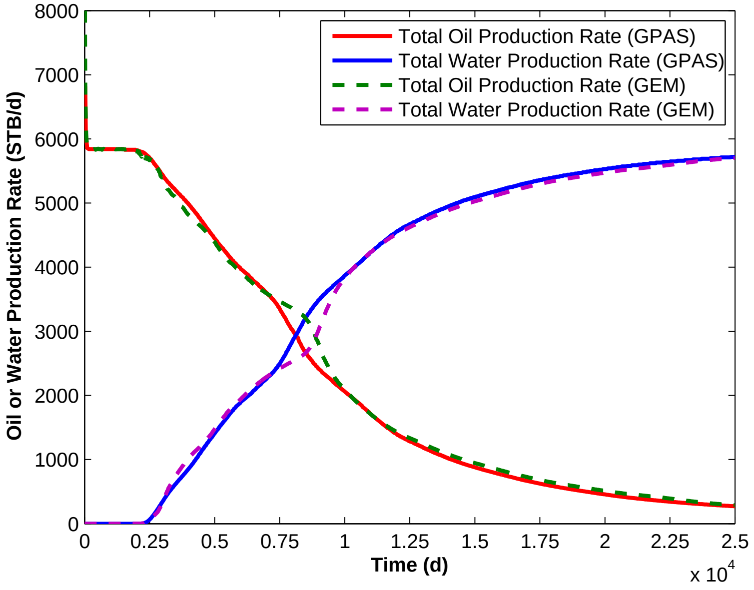


Figure 7.18: Total oil and water production rates for corner point case, Case 7.2

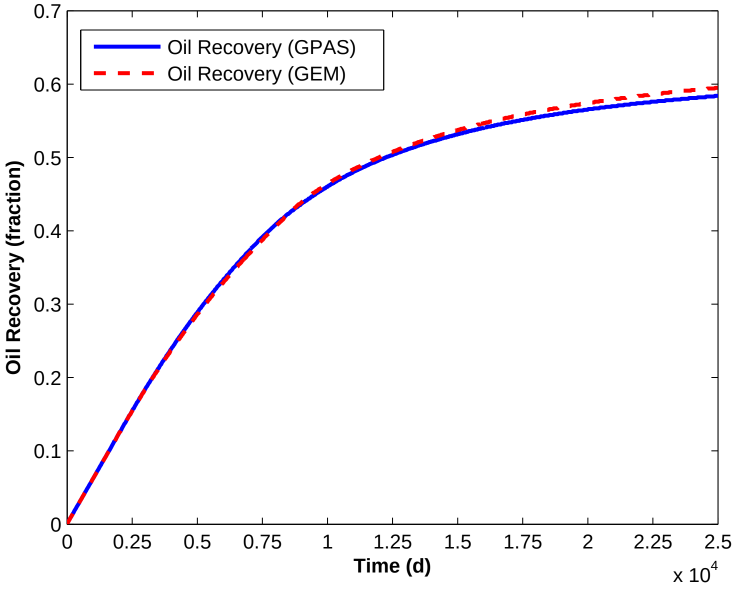


Figure 7.19: Field oil recovery for corner point case, Case 7.2

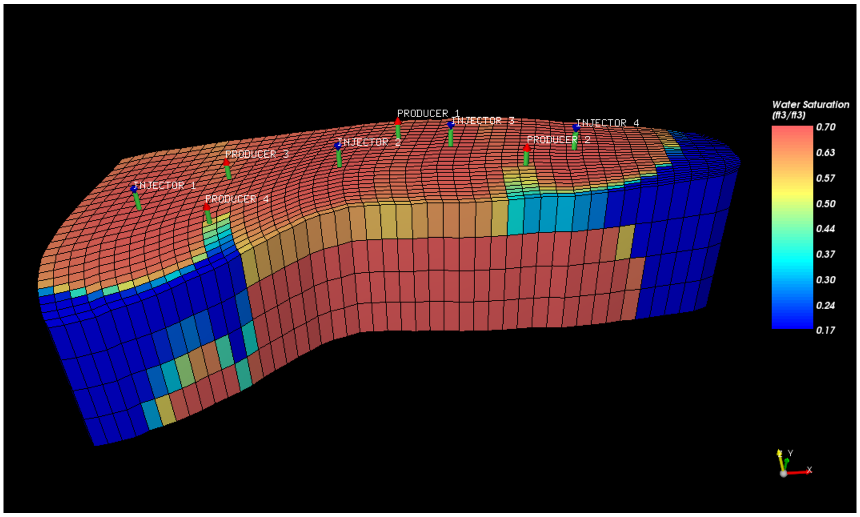


Figure 7.20: GPAS Saturation map at the end of simulation for corner point case, Case 7.2

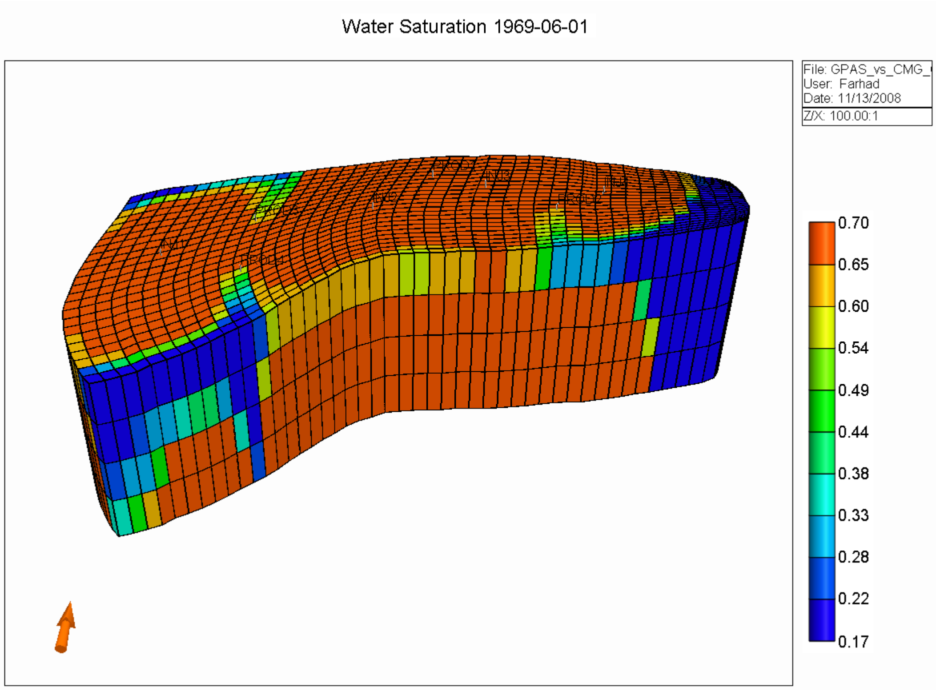


Figure 7.21: CMG Saturation map at the end of simulation for corner point case, Case 7.2

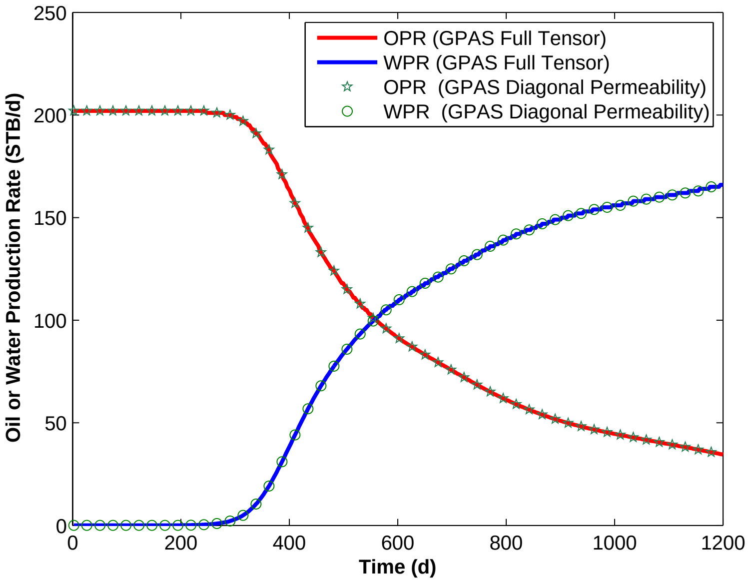


Figure 7.22: GPAS full permeability tensor vs. diagonal permeability options for a 3D waterflood case

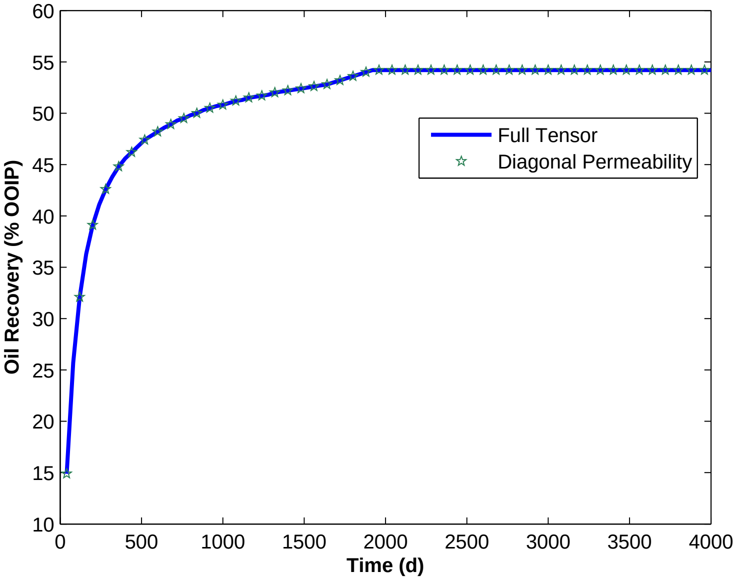


Figure 7.23: GPAS full permeability tensor vs. diagonal permeability options for a 2D surfactant flood case

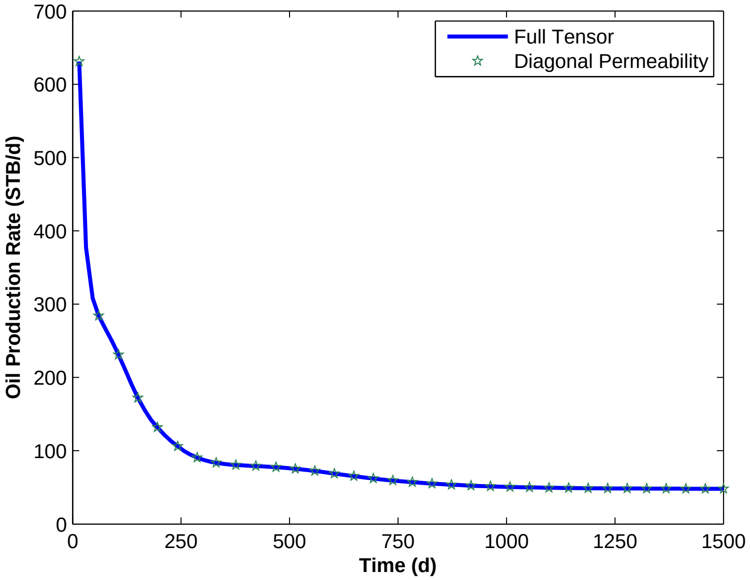


Figure 7.24: GPAS full permeability tensor vs. diagonal permeability options for the gas injection case

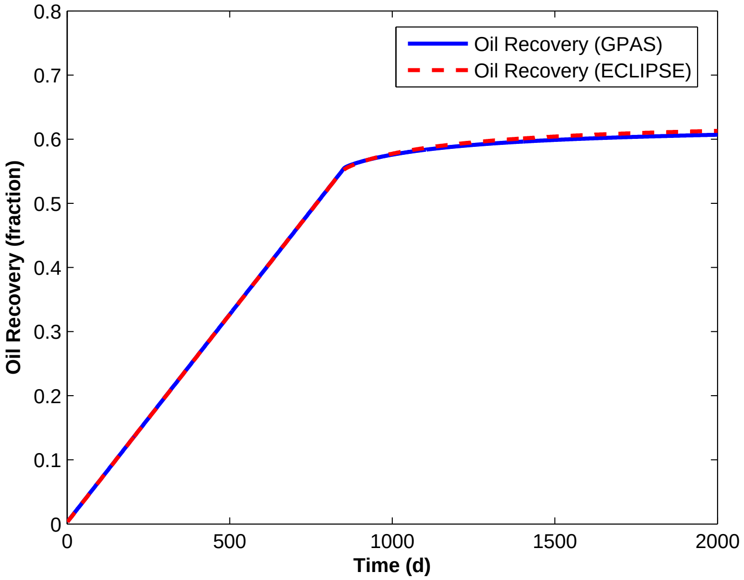


Figure 7.25: Recovery vs. time for the full tensor comparison between GPAS and ECLIPSE in 2D

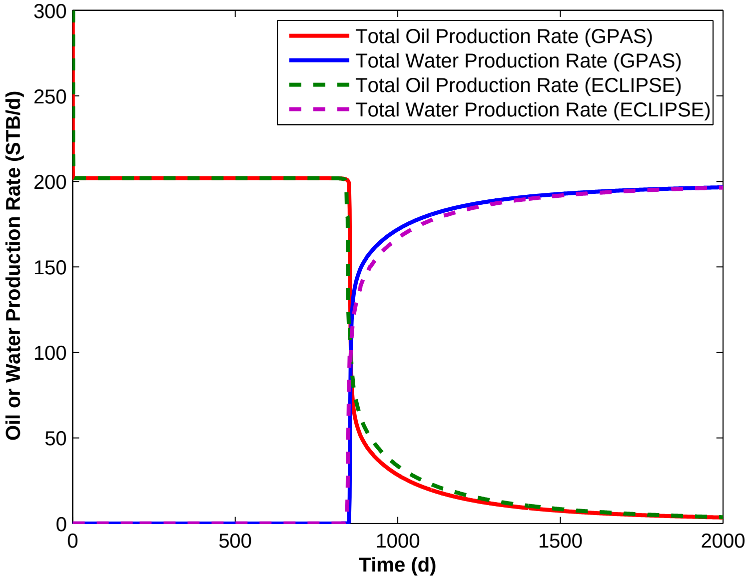


Figure 7.26: Field oil and water production rates for the full tensor comparison between GPAS and ECLIPSE in 2D

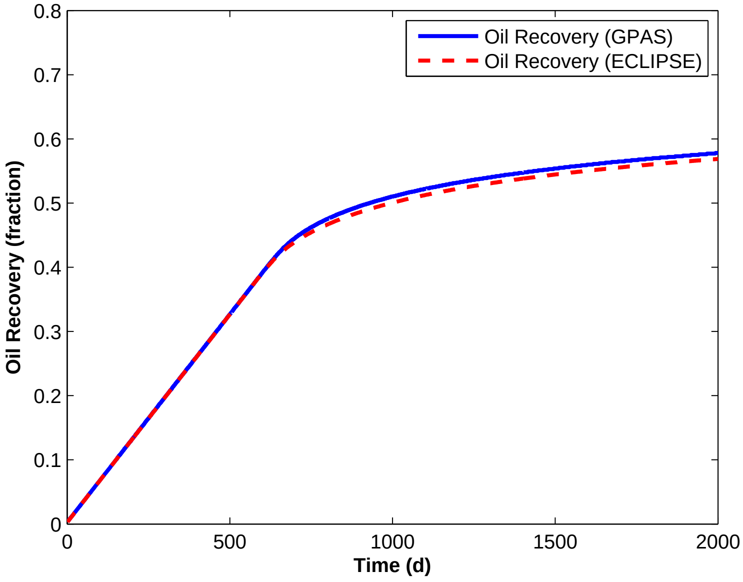


Figure 7.27: Recovery vs. time for the full tensor comparison between GPAS and ECLIPSE in 3D

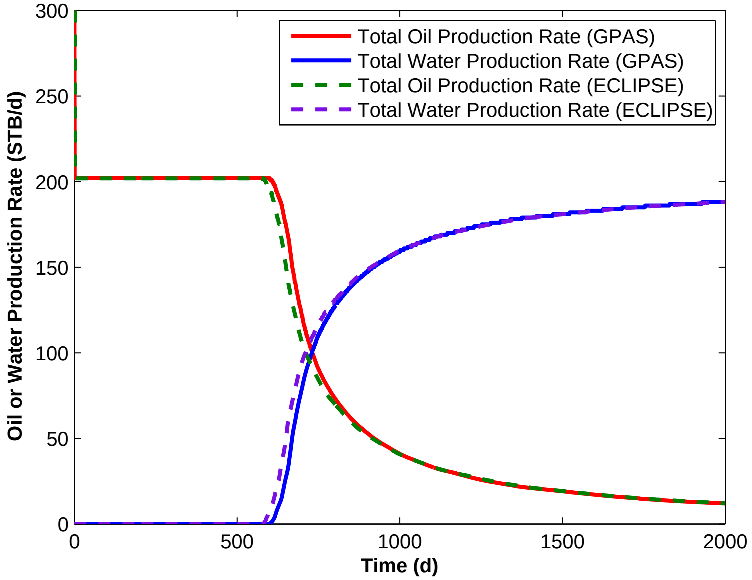


Figure 7.28: Field oil and water production rates for the full tensor comparison between GPAS and ECLIPSE in 3D

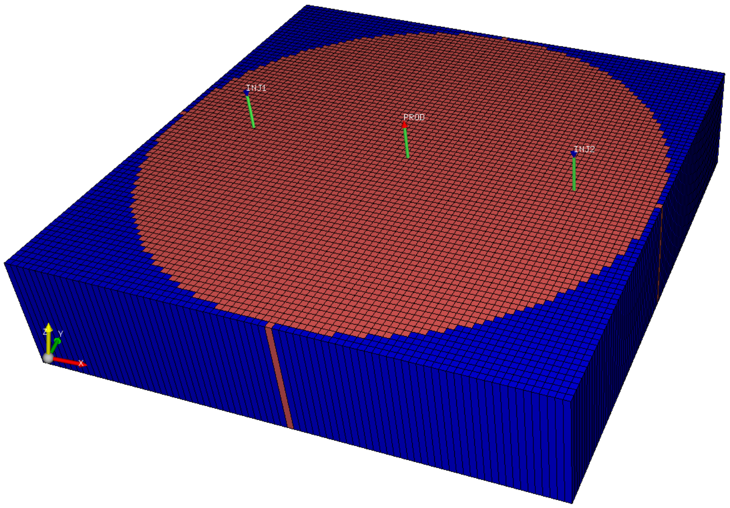


Figure 7.29: Original 2D cylindrical system used for Case 7.3.3

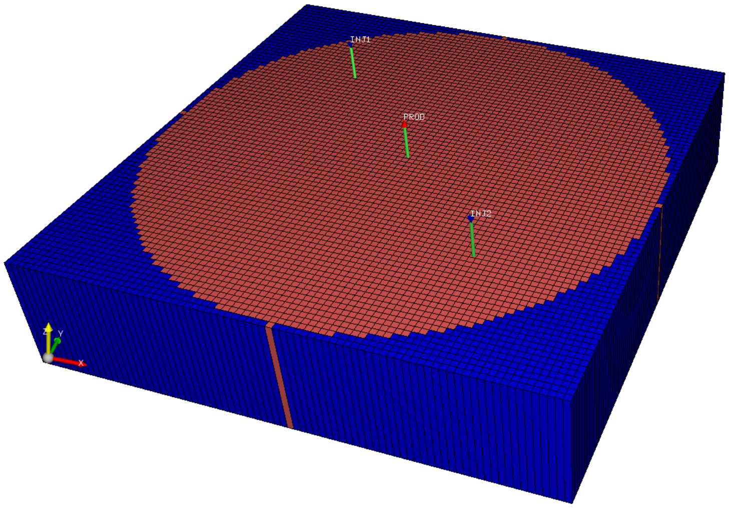


Figure 7.30: 2D cylindrical system after 45 degree rotation for Case 7.3.3

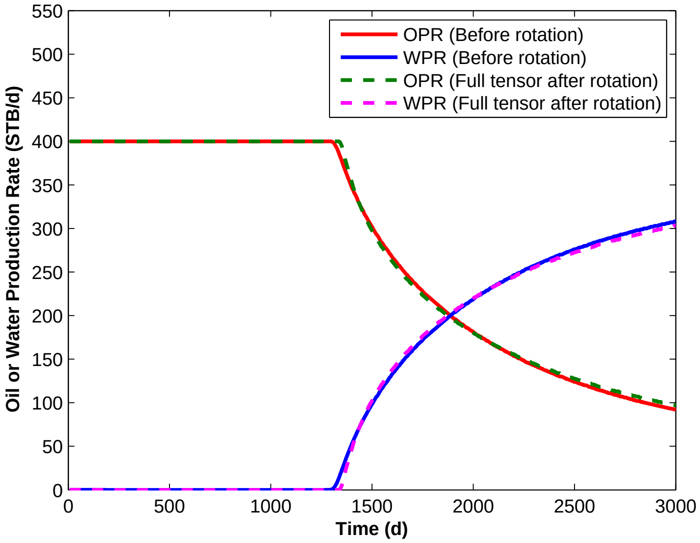


Figure 7.31: Oil and water production rates for the 2D cylindrical case before and after rotation (Case 7.3.3)

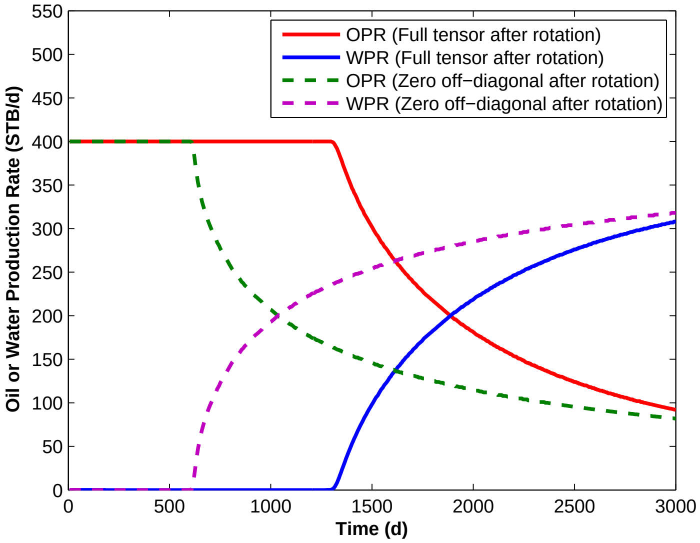


Figure 7.32: Effect of off-diagonal terms of the permeability tensor on the oil and water production rates (Case 7.3.3)

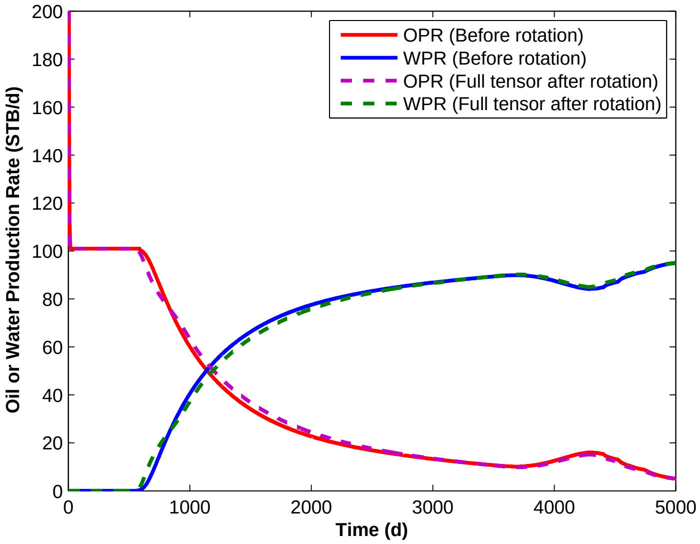


Figure 7.33: Oil and water production rates for the 2D cylindrical case before and after rotation for the chemical flood run (Case 7.3.3)

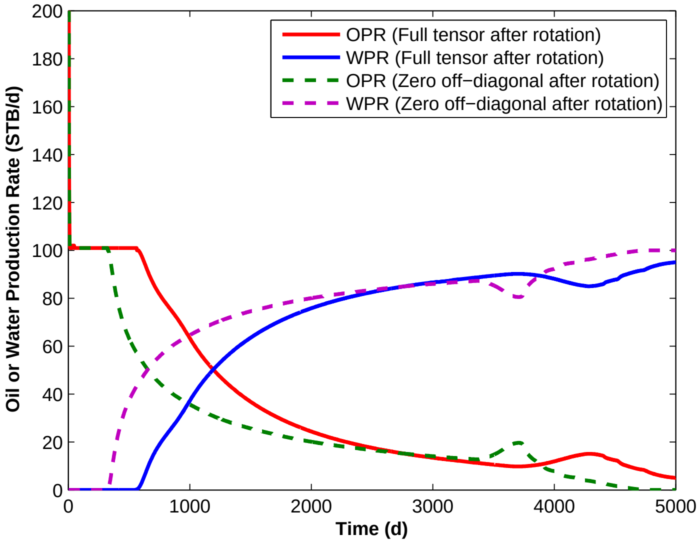


Figure 7.34: Effect of off-diagonal terms of the permeability tensor on the oil and water production rates for the chemical flood run (Case 7.3.3)

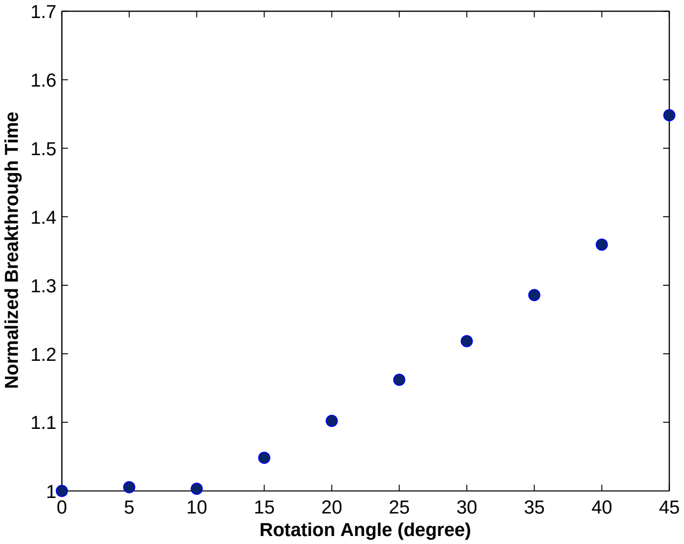


Figure 7.35: Effect of rotation angle on the water breakthrough time (Case 7.3.4)

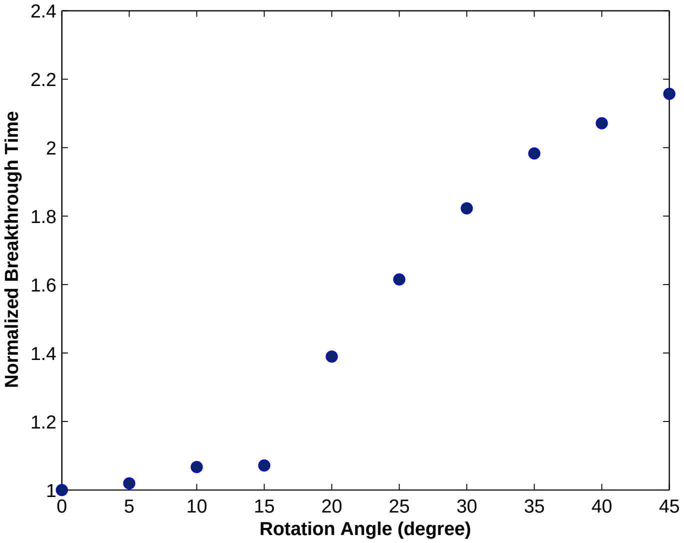


Figure 7.36: Effect of off-diagonal terms of permeability tensor vs. rotation angle (Case 7.3.4)

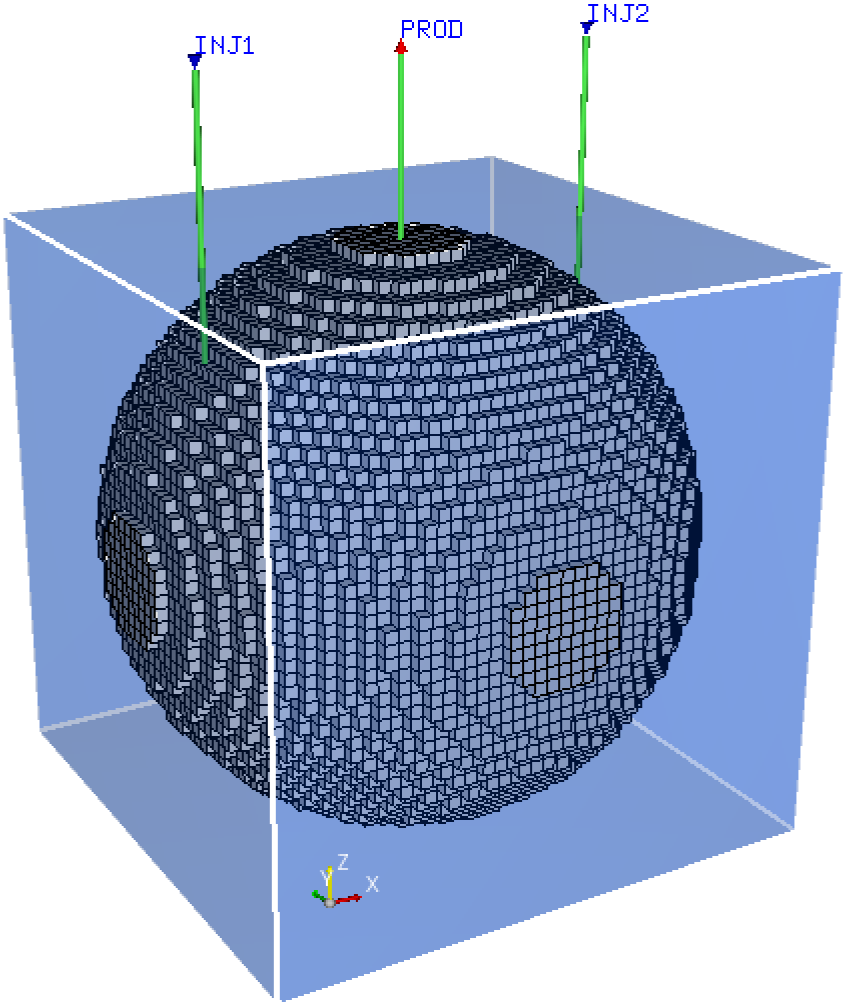


Figure 7.37: Schematic of the spherical reservoir (Case 7.3.5)

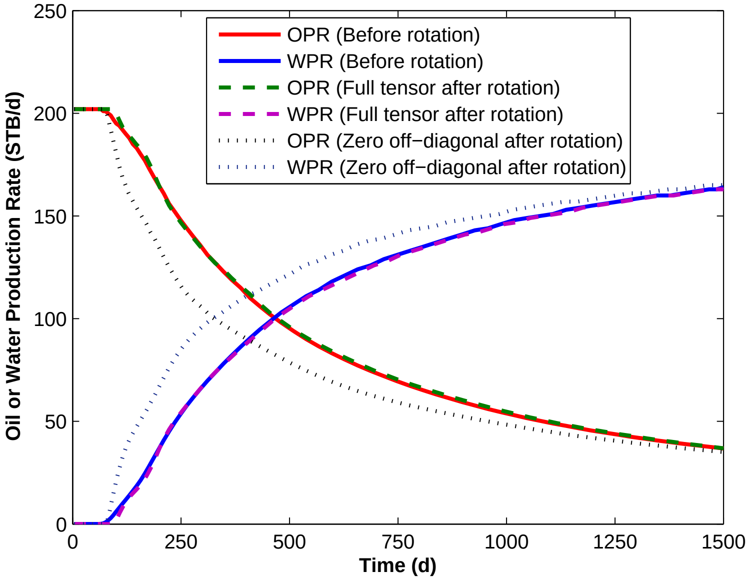


Figure 7.38: Oil and water rates for a spherical reservoir before and after rotation (Case 7.3.5)

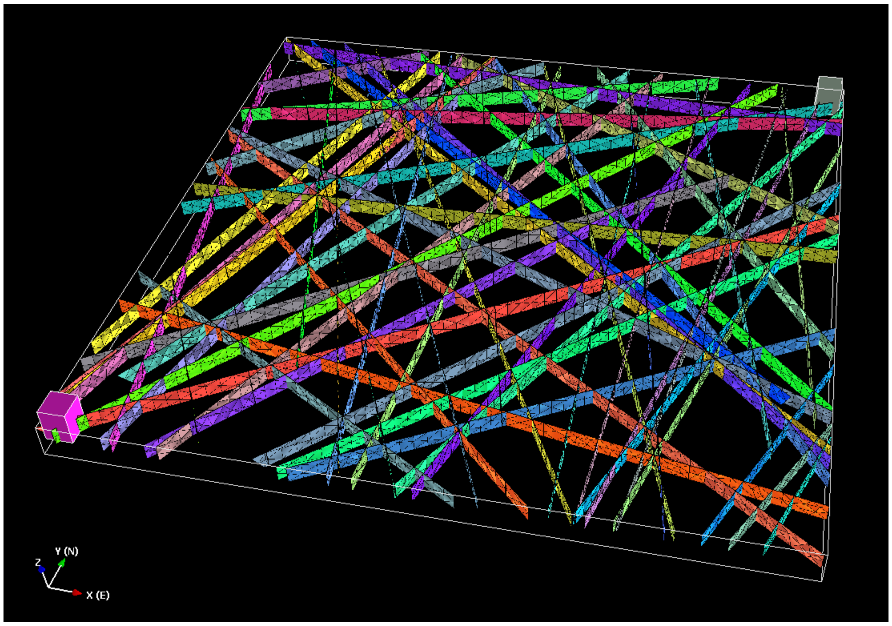


Figure 7.39: Schematic of the fracture system generated by FracMan for Case 7.3.6

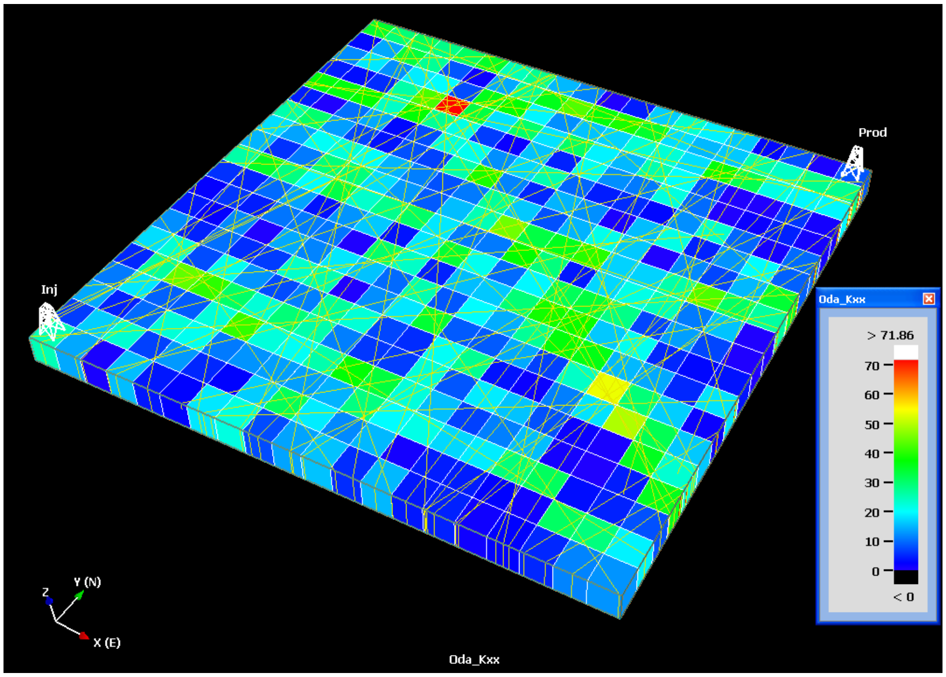


Figure 7.40: Equivalent permeability tensor for Case 7.3.6

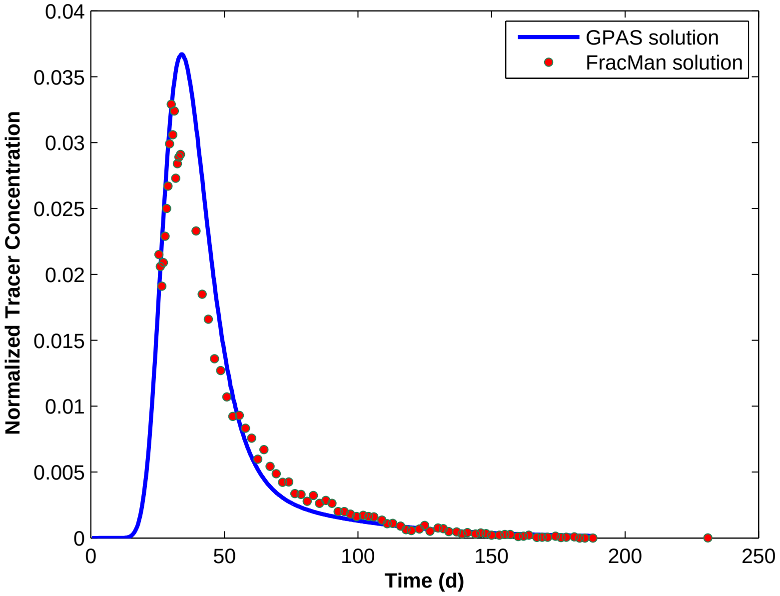


Figure 7.41: Tracer concentration vs. time for GPAS and FracMan for Case 7.3.6

Chapter 8

Parallel Processing

Parallel processing refers to the concept of dividing a program into multiple fragments such that each fragment can execute on a single processor. GPAS has been developed based on a framework approach. In this approach, the framework, named IPARS (Parashar *et al.*, 1997; Wang *et al.*, 1997), provides an umbrella for different physical models. The framework separates the physical model form the parallel processing models. The performance and scalability of the developed dual porosity models and full permeability tensor implementation are presented in this chapter.

One way to measure the parallel processing performance is the speedup test defined by

$$S_p = \frac{T_1}{T_p} \tag{8.1}$$

where T_1 is the execution time of the sequential algorithm on a single processor and T_p is the execution time of the parallel algorithm with p processors. A linear speedup or ideal speedup is obtained when $S_p = p$. The efficiency of a parallel run is measured by

$$E_p = \frac{S_p}{p} = \frac{T_1}{pT_p} \tag{8.2}$$

The efficiency is between zero and one, which is an indication of the time spent on solving the problem compared to the time spent on communication and synchronization.

GPAS uses a domain decomposition method for parallel processing. In a domain decomposition method, a boundary value problem splits into smaller boundary value problems such that each problem can be solved in a local machine while the boundary conditions are being updated during the runtime. The domain decomposition in GPAS is done along the y direction.

The Lonestar cluster of the Texas Advanced Computing Center (TACC) which is one of the largest academic computational resources in the nation was used for performing parallel processing case studies using GPAS. The Lonestar Linux Cluster has 1300 nodes, with 2 Dual-Core processors per node, for a total of 5200 cores. Each compute node consists of a Dell PowerEdge 1955 blade. Each node contains two Xeon Intel Duo-Core 64-bit processors (4 cores per node) on a single board. The Core frequency is 2.66GHz and each node contains 8GB of memory.

In the first case, we present the execution time and speedup of a chemical flood process for a 3D reservoir. In the second case, we study the scalability of GPAS in a highly heterogeneous fractured reservoir. Finally, the performance of a very large and heterogeneous system for a waterflood flood with full permeability tensor is presented.

8.1 Chemical Dual Porosity Case Study

To test the consistency and speedup of the developed chemical dual porosity model, we considered a 3D model with four five-spot patterns. There are nine injectors and four producers in the system. The reservoir size is 1920x1920x50 ft ^{3}. The total number of gridblocks used in the simulation is 81,920. The fracture system of the reservoir is discretized into a 64x64x5 (20,480) gridblocks where within each gridblock there is a matrix block with 2x2 subgrids. Figure 8.1 shows a schematic of the reservoir and well locations. The fracture system has a constant porosity of 0.02 and a constant permeability of 500 md. The matrix system has a porosity of 0.18 and a permeability of 5 md. Table 8.1 gives reservoir and fluid properties. The injectors are operating at a constant rate of 100 $\frac{STB}{d}$ and the producers are operating under a constant bottomhole flowing pressure of 3900 $psia$. A waterflood is performed for the first 1000 days following a chemical flood with a surfactant concentration of 0.05 volume fraction for the next 1000 days.

The oil rate and surfactant concentration for well P1 is shown in Figs. 8.2 and 8.3, respectively. The results are identical regardless of the number of processors. Figures 8.4 and 8.5 show the field oil and water production rates. The results show the consistency between parallel runs. Table 8.2 shows the execution time and speedups for this case. The execution time and speedup are shown in Figs. 8.6 and 8.7 for the different number of processors.

In this particular case, the speedup of 14.1 is obtained using 16 processors. Whereas, the speedup of 23.8 was achieved using 32 processors. More time is spent for communication and synchronization between processors when the number of processors increases. This is due to the fact that each computational portion of the reservoir becomes smaller for each processor as the number of processors increases, therefore, the time spent on message passing and communication becomes dominant. To address this issue, the CPU time for the major parts of the code such as linear solver, updating Jacobian, matrix linear solver, and updating matrix Jacobian are plotted in Figs. 8.8 and 8.9. The parallel linear solver which is responsible for most of the communication between processors takes a large amount of CPU time using 32 processors (47%) compared to 2 processor runs (32%). Note that all matrix calculations are done locally in each processor such that they do not affect the parallel scalability of GPAS.

8.2 Heterogeneous Waterflood Case Study

In this case study, a highly heterogeneous reservoir is used to test the parallel scalability of GPAS. The reservoir has 960 ft in length, 960 ft in width, and 40 ft in thickness. The fracture system is divided into 32x32x4 gridblocks and the matrix system is divided into 2x2 subgrids. A Dykstra-Parson coefficient of 0.7 was used to generate a permeability field in the x direction using a Sequential Gaussian Simulation program. The y permeability is assumed to be the same as the permeability in the x direction and the permeability in the z direction is set to be 10% of the permeability in the x direction. The matrix system also has a heterogenous permeability and porosity field. Figures 8.10 and 8.11 show the permeability maps for fracture and matrix system, respectively. Figure 8.12 shows the porosity map for the matrix system.

There are four injectors in the system which inject 600 $\frac{STB}{d}$ and there is one producer operating at a constant bottomhole flowing pressure of 3900 $psia$. The initial reservoir pressure is 4000 $psia$. A waterflood is performed for 1500 days using 1, 2, 4, 8, and 16 processors. To ensure the consistency between results, the total oil and water production rates for different number of processors are shown in Fig. 8.13. There is a very good agreement between the results. Figures 8.14 and 8.15 show the average water saturations for the fracture and matrix system at the end of the simulation, respectively. The execution time and parallel speedup for this case are shown in Figs. 8.16 and 8.17. A good speedup is obtained until eight processors, but reduces after that. This happens because the problem size for each processor becomes smaller such that more time is spent in communication between processors.

8.3 Full Permeability Tensor Case Study

To illustrate the performance of the implemented full permeability tensor in a parallel processing platform, a very large 3D waterflood model with 9 injectors and 4 producers was performed. The total number of gridblocks in this case is 1,310,720. The fracture system is discretized in a 256x256x5 gridblock system and the matrix blocks are discretized into 2x2 subgrids. This leads to 5,242,880 unknowns in the fracture and matrix media per each newton iteration. The fracture system has a highly heterogeneous permeability field. FracMan/FracWorks (Golder Associates, 2007) was used to generate a fractured system and the generated fractured system was converted to an equivalent full permeability tensor for each gridblock using Oda's method (Oda, 1985). A schematic of the permeability map for this system is shown in Fig. 8.18. A closer look at the permeability map shows that the equivalent permeability tensor can capture the fracture heterogeneity and anisotropy (Fig. 8.19). The matrix system has a constant permeability of 1 md. Well patterns and flow rates are the same as those used in the chemical flood run (Case 8.1).

Due to the large problem size and limitation on computational resources, a waterflood was performed for only 365 days using 2, 4, 8, 16, 32, 64, 128 processors. To verify the consistency of the results, the water and oil production rates for production wells were compared. Figures 8.20 and 8.21 show the oil and water production rates for well P1. The results are almost identical regardless of the number of processors. The total oil and water production rates for the field are shown in Figs. 8.22 and 8.23. The figures show excellent agreement between the results for different numbers of processors.

In Figs. 8.24 and 8.25 the execution times and speedup are presented. The speedup is very good until 32 processors, but it reduces after that.

Table 8.1: Input parameters for parallel chemical injection for Case 8.1

Description	Fracture	Matrix
Number of gridblocks	64x64x5	2x2
Porosity	0.02	0.18
Permeability	$K_{xx} = 500$ md $K_{yy} = 500$ md $K_{zz} = 50$ md	$K_{xx} = 5$ md $K_{yy} = 5$ md $K_{zz} = 5$ md
Initial water saturation	0.1	0.25
Water viscosity	0.5 cp	0.5 cp
Heights of bimodal curve	0.07, 0.04 volume fraction	
Salinity limits	0.177, 0.25 $\frac{meq}{ml}$	
CMC	0.0001	
Adsorption parameters	1.5, 0.5, 1000	
IFT parameters	9.0, 2.0	
Low trapping number relative permeability	Residual saturations: 0.10 0.001 0.1 Exponents : 1.46 2.15 1.0 Endpoints : 1.00 1.00 1.0	0.25 0.30 0.25 1.18 1.80 1.18 0.20 0.92 0.20
High trapping number relative permeability	Residual saturations: Exponents Endpoints	0.00 0.00 0.00 1.18 1.80 1.18 1.00 1.00 1.00
Initial reservoir pressure	4000 psia	
Injection rate	900.0 $\frac{STB}{d}$	
Production well pressure	3900 psia	

Table 8.2: Execution times and speedups for Case 8.1

Number of Processors	Execution time (sec)	Speedup
2	64252.56	2
4	34731.15	3.7
8	18897.81	6.8
16	9113.84	14.1
32	5406.408	23.8

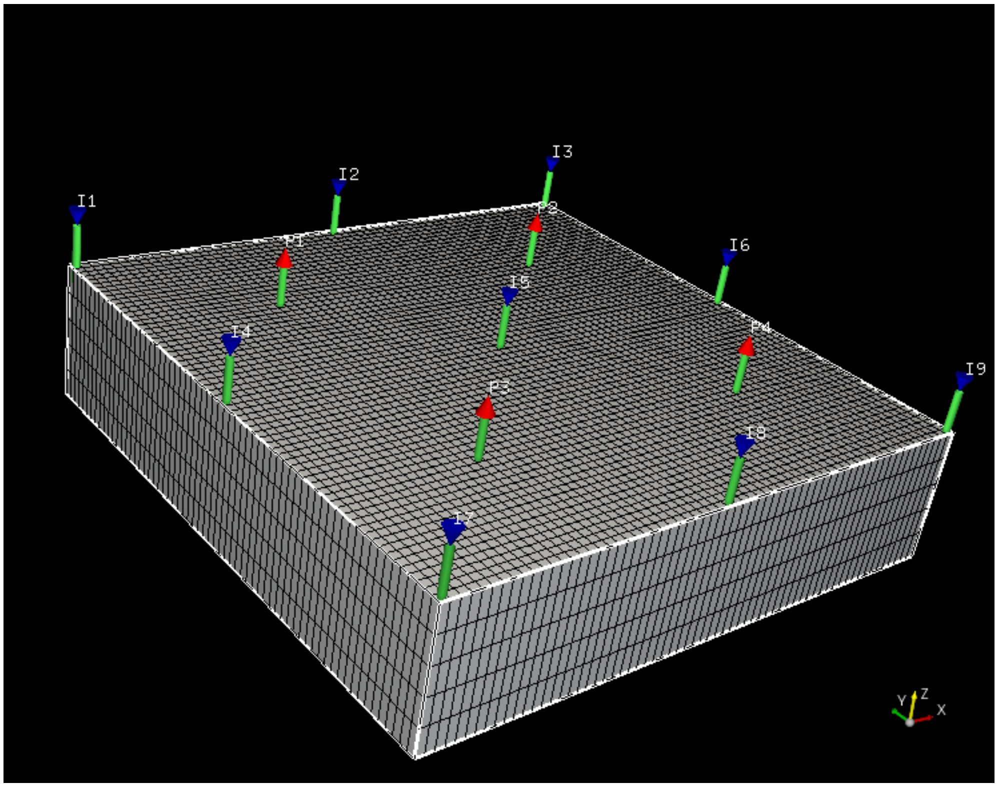


Figure 8.1: Schematic of the reservoir used for Case 8.1

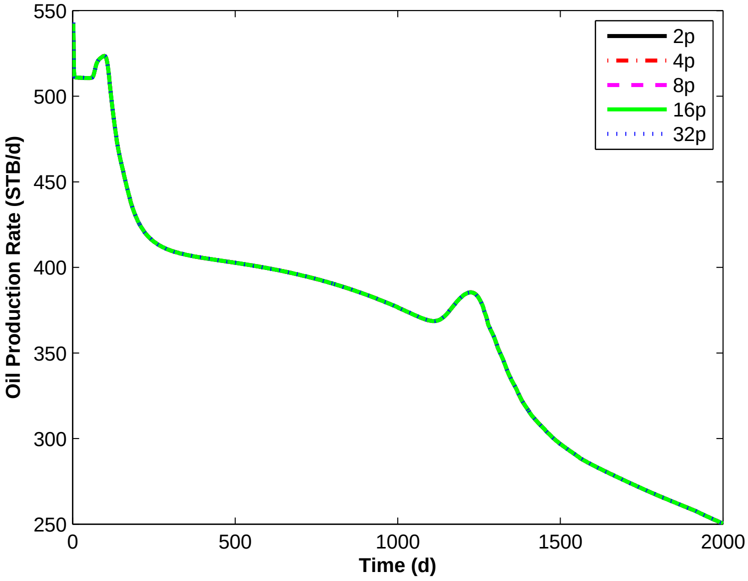


Figure 8.2: Oil production rate of well P1 for Case 8.1

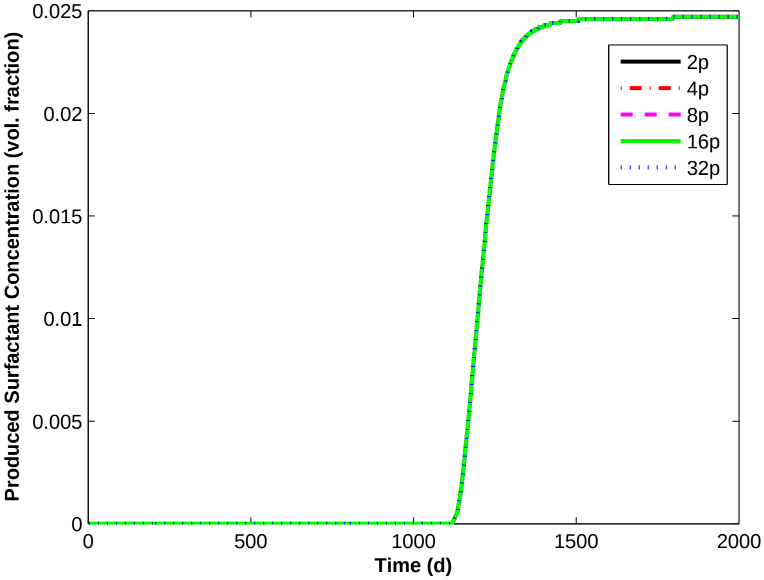


Figure 8.3: Produced surfactant concentration of well P1 for Case 8.1

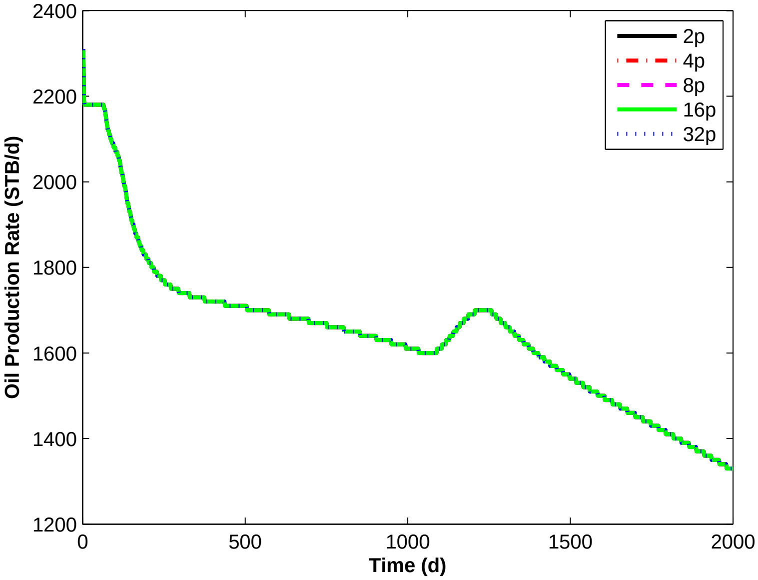


Figure 8.4: Field oil production rate for Case 8.1

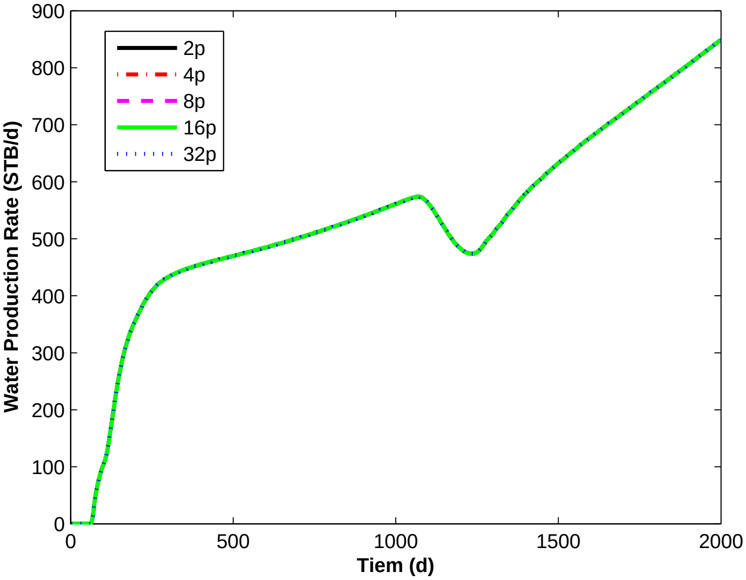


Figure 8.5: Field water production rate for Case 8.1

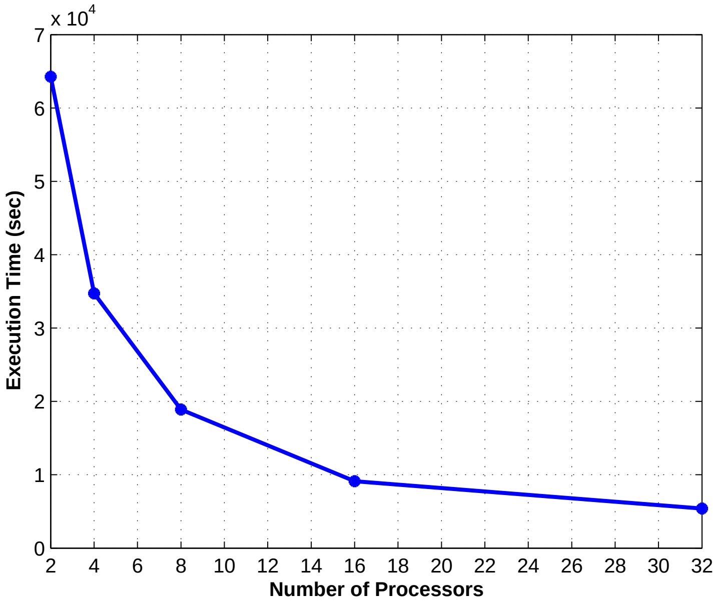


Figure 8.6: Execution time for Case 8.1

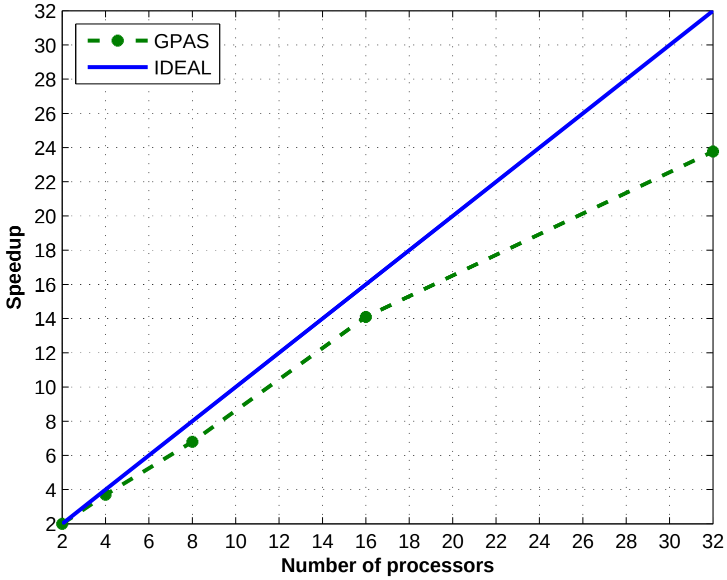


Figure 8.7: Speedup for Case 8.1

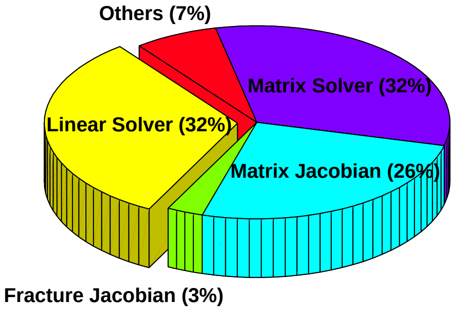


Figure 8.8: Execution time breakdown in 2 processors for Case 8.1

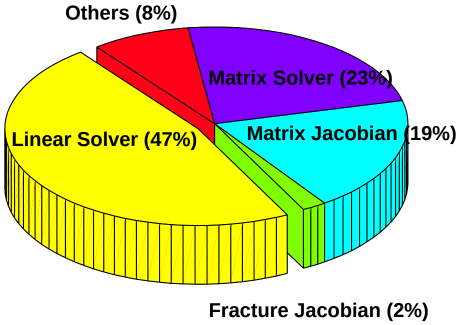


Figure 8.9: Execution time breakdown in 32 processors for Case 8.1

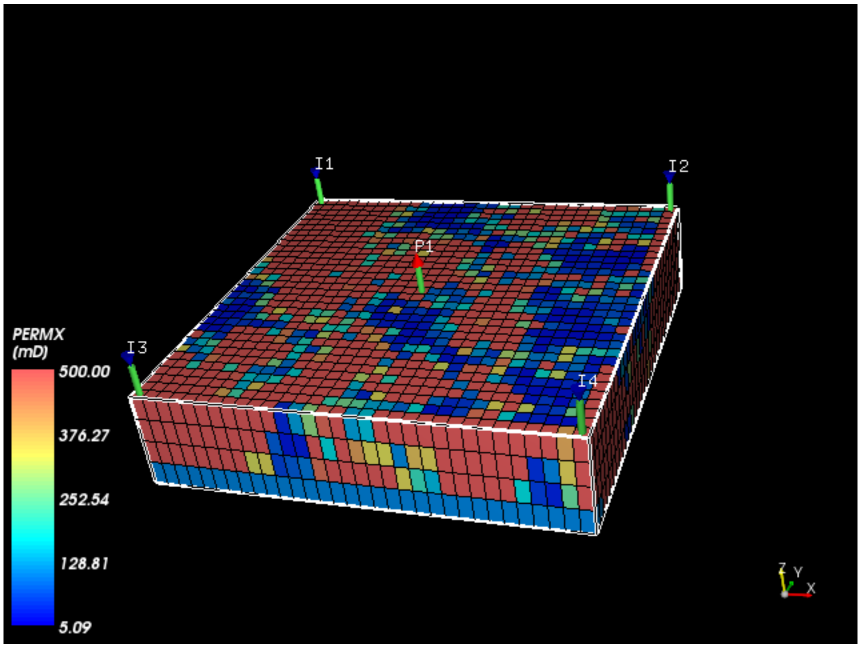


Figure 8.10: The permeability map for the fracture system in Case 8.2

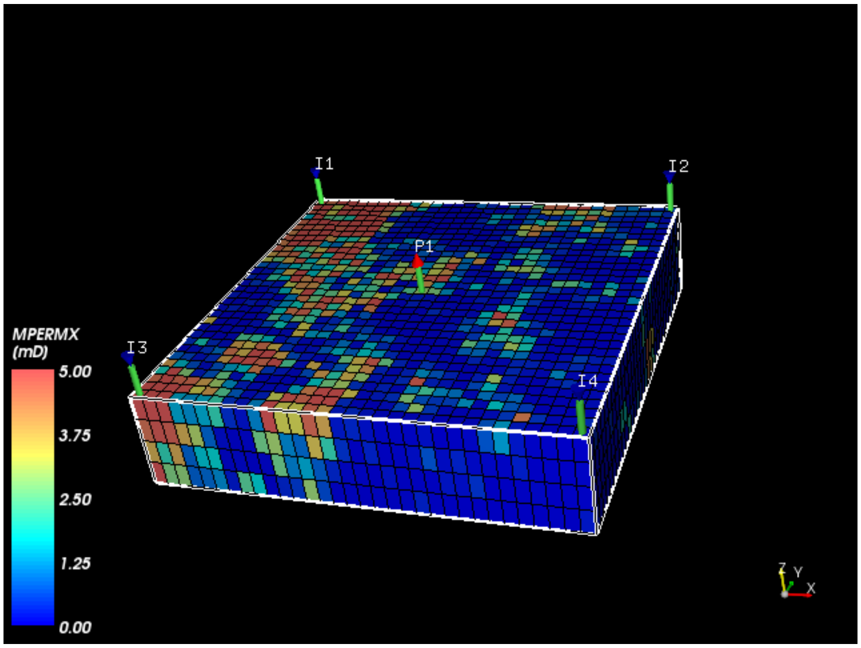


Figure 8.11: The permeability map for the matrix system in Case 8.2

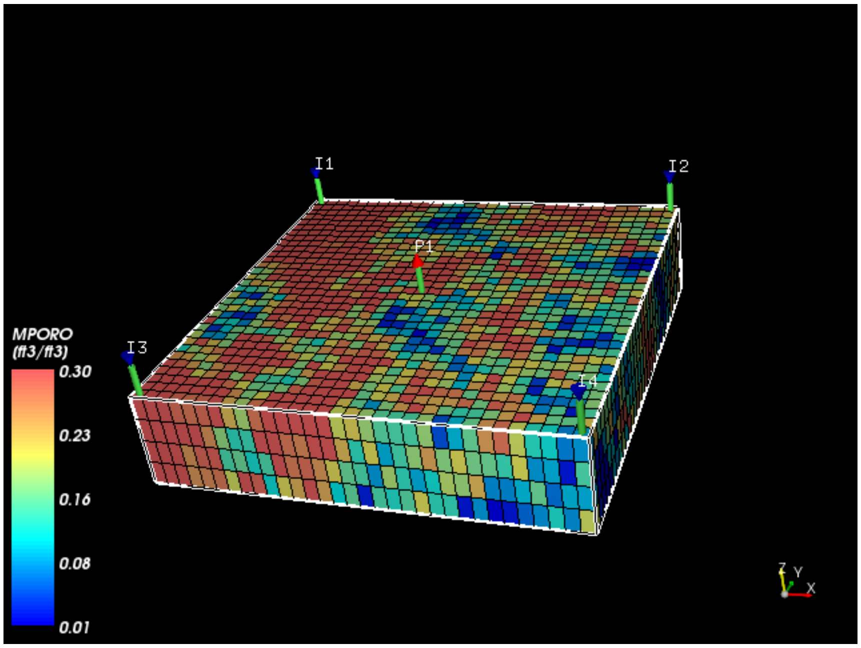


Figure 8.12: The porosity map for the matrix system in Case 8.2

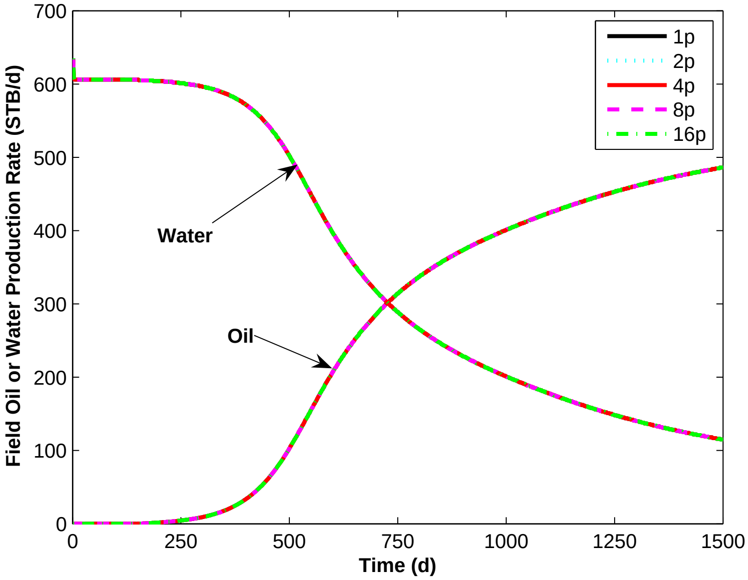


Figure 8.13: Field oil and water rates for Case 8.2

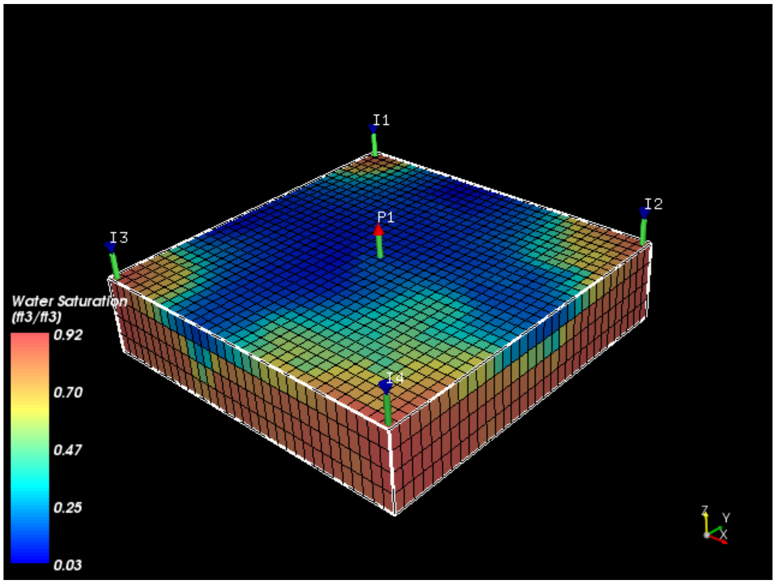


Figure 8.14: Water saturation map for the fracture system at the end of simulation for Case 8.2

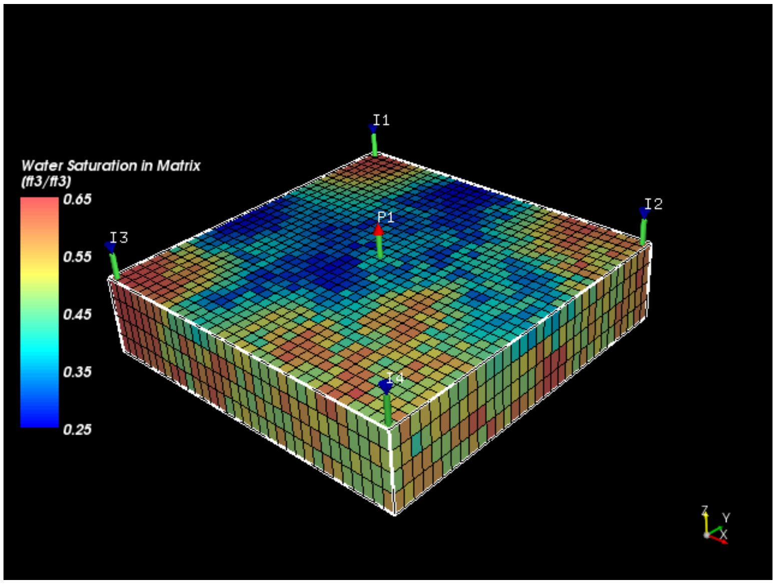


Figure 8.15: Water saturation map for the matrix system at the end of simulation for Case 8.2

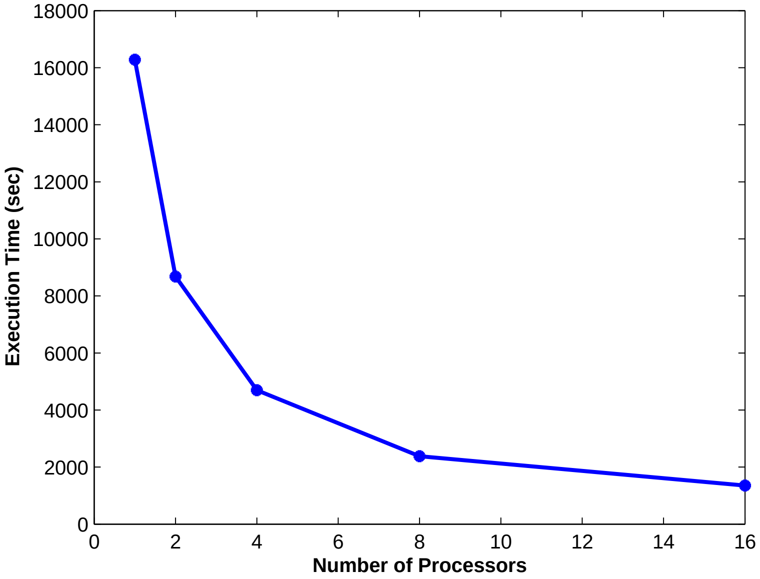


Figure 8.16: Execution time for Case 8.2

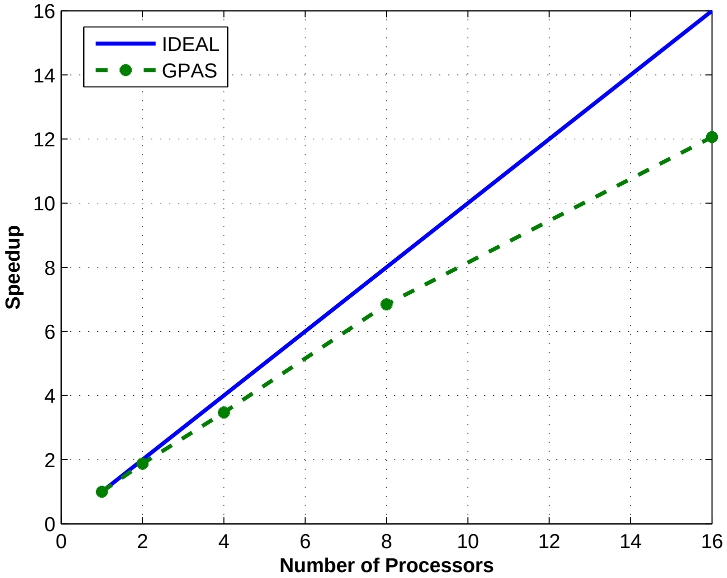


Figure 8.17: Speedup for Case 8.2

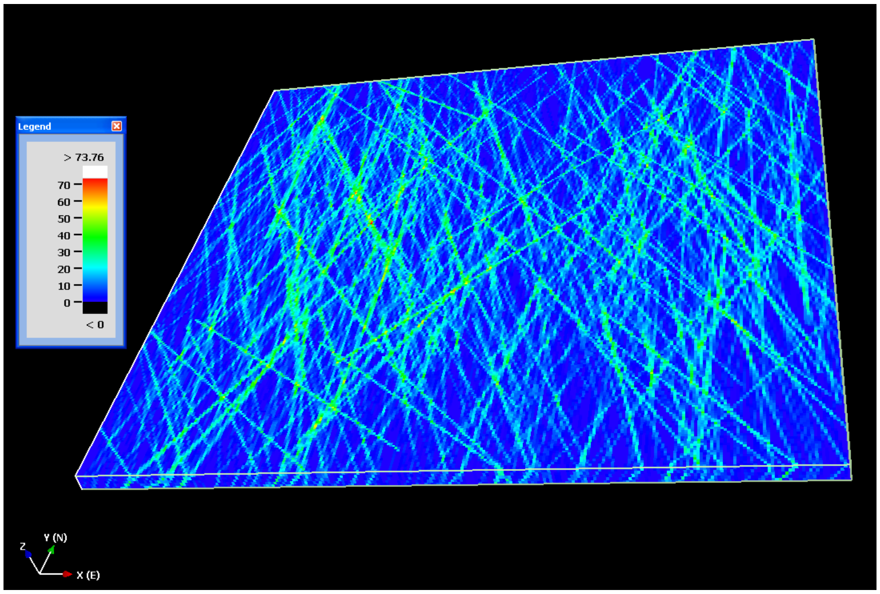


Figure 8.18: Permeability map for Case 8.3

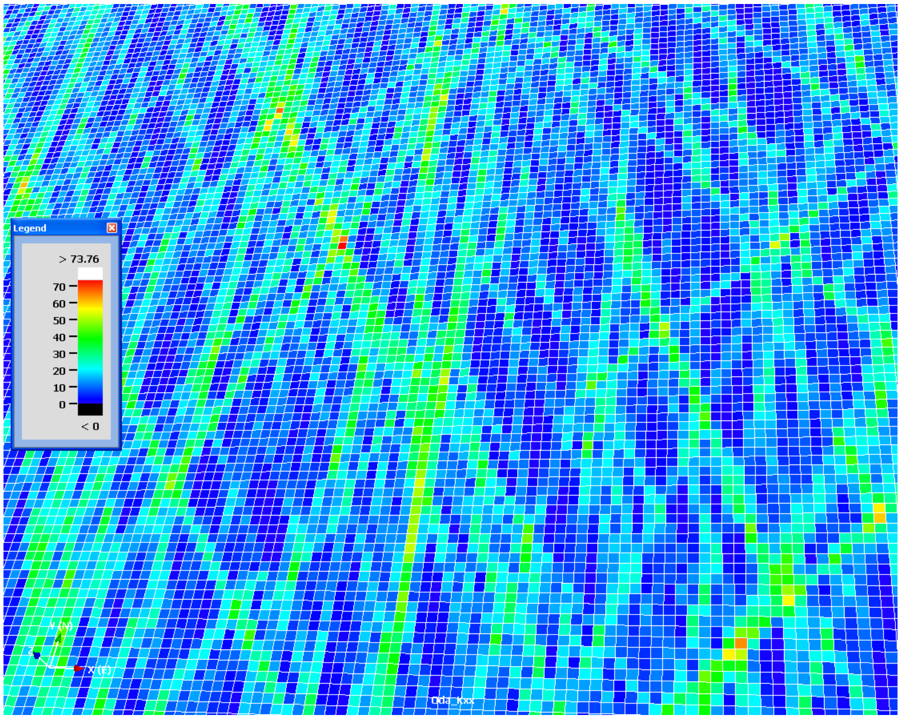


Figure 8.19: Fracture gridblock permeabilities for Case 8.3

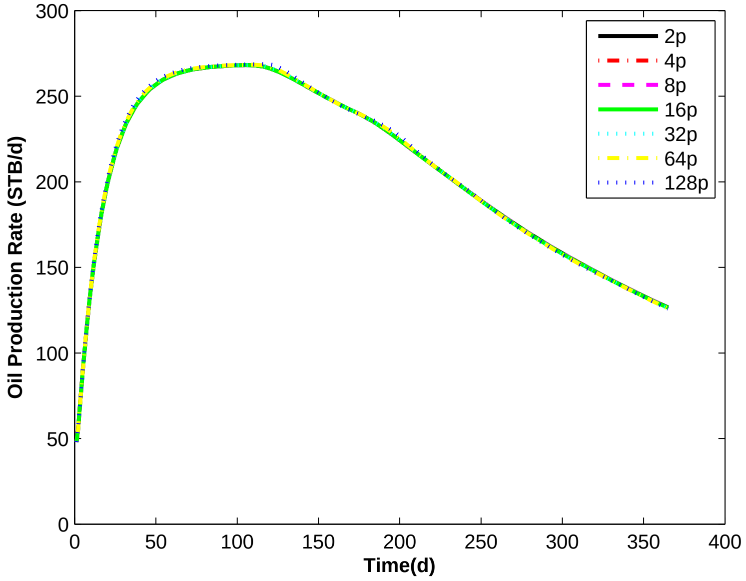


Figure 8.20: Oil production rate for well P1 for Case 8.3

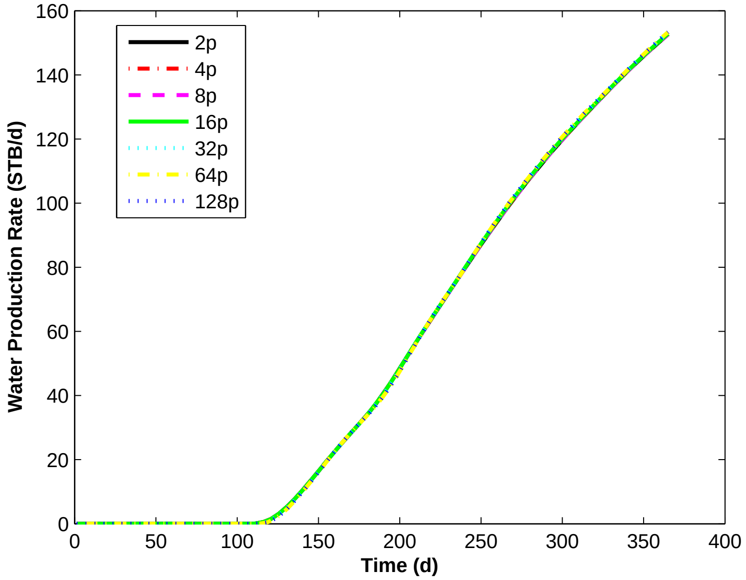


Figure 8.21: Water production rate for well P1 for Case 8.3

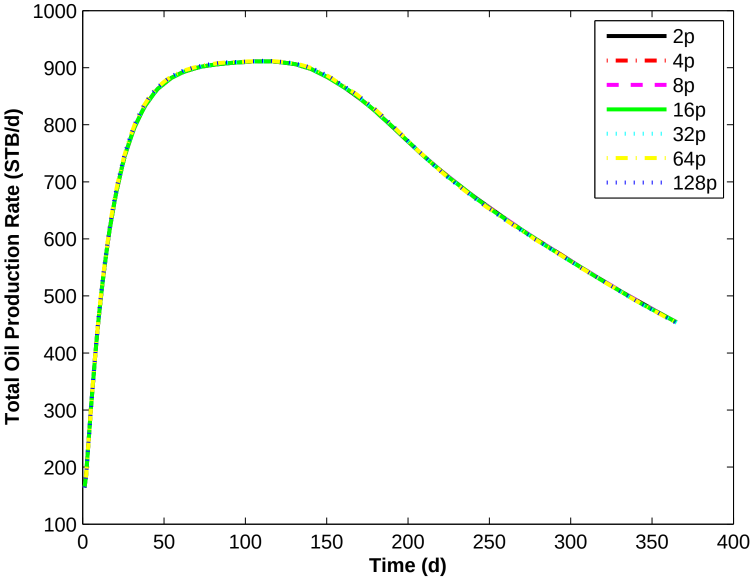


Figure 8.22: Total oil production rate for Case 8.3

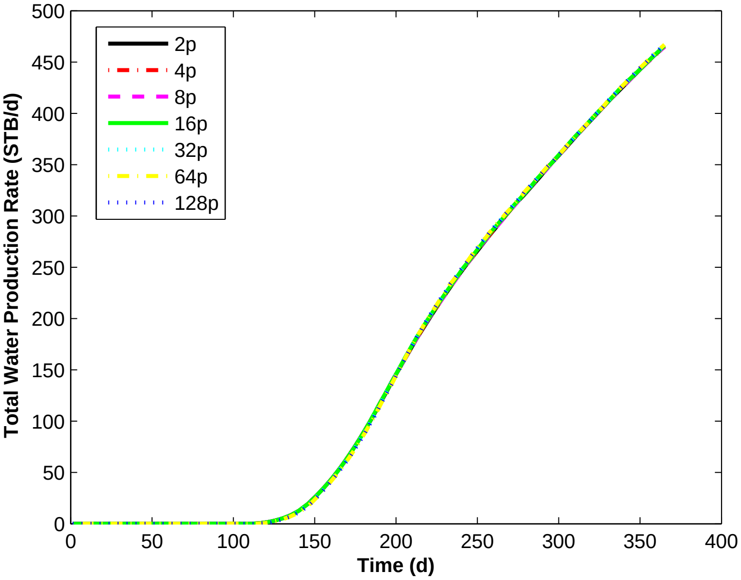


Figure 8.23: Total water production rate for Case 8.3

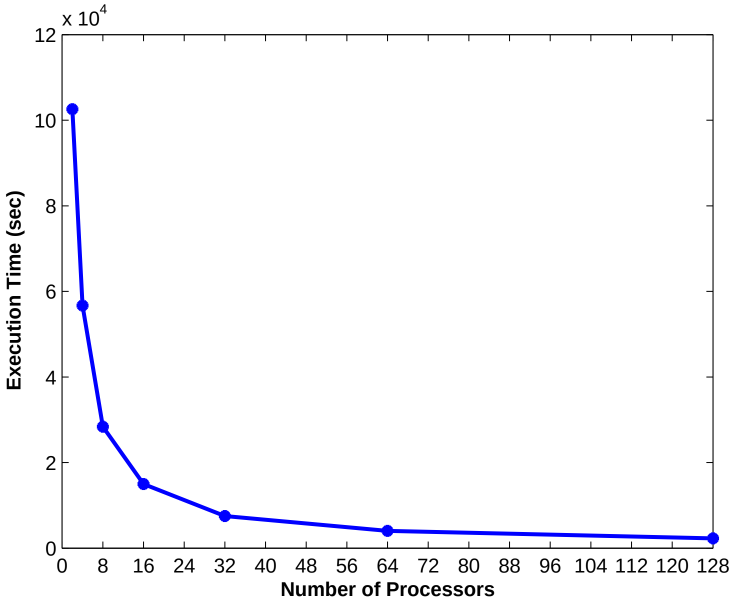


Figure 8.24: Execution time for Case 8.3

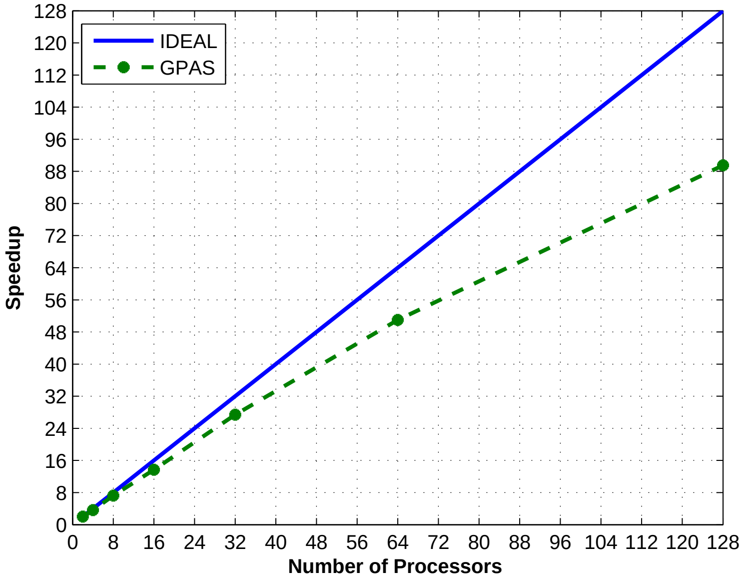


Figure 8.25: Speedup for Case 8.3

Chapter 9

Summary, Conclusions and Recommendations

9.1 Summary

1. A new, parallel, chemical, fully implicit dual porosity simulator was developed to simulate naturally fractured reservoirs. The model was verified against the UTCHEM simulator. Also, for compatibility with the new development for the full permeability tensor, the EOS compositional dual porosity model of GPAS was rewritten and verified against the CMG commercial simulator.

2. A multiple interacting continua concept was used to discretize the matrix blocks in the horizontal direction and a stacked grid concept was used to discretize the matrix blocks in the vertical direction.

3. A corner point method was implemented in GPAS and was verified against the CMG commercial simulator. The results showed excellent agreement between the two simulators.

4. A full tensor representation of permeability was implemented in both chem-

ical and EOS compositional modules of GPAS. The implementation yields a 19-point stencil scheme in 3D. The implementation was verified against the ECLIPSE commercial simulator. Also, verifications were performed by comparing the results of the full permeability tensor implementation with a diagonal tensor model, and by studies on cylindrical and spherical reservoirs.

5. The consistency and efficiency of the parallel processing features of the developed modules in GPAS were verified using a chemical flood, a waterflood, and a waterflood with full permeability tensor simulations.

9.2 Conclusions

The following conclusions are drawn from this study:

1. Using a diagonal permeability tensor is not adequate to capture the heterogeneity and anisotropy of naturally fractured reservoirs.
2. When the principle directions of a permeability tensor are not aligned with the simulation coordinate system, a full permeability tensor results. Our studies show that when the simulation coordinate system is rotated by angles of 20° or higher, for an anisotropy ratio of 1:10, using the original diagonal permeability tensor to perform the simulations, water breakthrough time errors of 16% or higher (depending on the angle of rotation) are observed.
3. Ignoring the off-diagonal terms of the permeability tensor leads to large errors, especially when the anisotropy is large. This study shows that the off-diagonal terms are very important in naturally fractured reservoirs where heterogeneity and anisotropy are large.
4. Our full tensor dual porosity simulator gives close results to the results of a discrete fracture simulator.

9.3 Recommendations

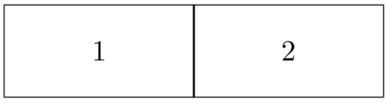
The following recommendations are suggested for future study.

1. Investigation needs to be carried out to show when the dual porosity model with diagonal permeability tensor is sufficient enough to model a fractured system and when a full permeability tensor model is needed.
2. The results of the full permeability tensor model need to be verified against the results of a discrete fracture network model for multi-phase systems.
3. The effect of the grid refinement on the full permeability tensor results should be investigated.
4. The Oda's method should be extended for the system where the fractures are not well connected. The Oda's method assumes that all fractures are highly connected and all contribute to the overall flow of the system.
5. The developed dual porosity model should be further extended to account for the exchange of fluid between the matrix blocks (dual permeability model).

Appendix A

Constant Part of the Transmissibility

Darcy's law can be used to calculate the overall pressure drop in a system consisting two gridblocks with different permeabilities.



$$P_{1+\frac{1}{2}} - P_1 = \frac{\frac{\Delta X_1}{2} u\mu}{k_1} \tag{A.1}$$

$$P_2 - P_{1+\frac{1}{2}} = \frac{\frac{\Delta X_2}{2} u\mu}{k_2} \tag{A.2}$$

$$P_2 - P_1 = \frac{\left(\frac{\Delta X_1}{2} + \frac{\Delta X_2}{2}\right)}{k_{1+\frac{1}{2}}} \tag{A.3}$$

Since $P_2 - P_1 = \left(P_2 - P_{1+\frac{1}{2}}\right) + \left(P_1 - P_{1+\frac{1}{2}}\right)$, then

$$\frac{\Delta X_1 + \Delta X_2}{k_{1+\frac{1}{2}}} = \frac{\Delta X_1}{k_1} + \frac{\Delta X_2}{k_2} \tag{A.4}$$

Solving for $k_{1+\frac{1}{2}}$ yields

$$k_{1+\frac{1}{2}} = \frac{\Delta X_1 + \Delta X_2}{\frac{\Delta X_1}{k_1} + \frac{\Delta X_2}{k_2}} \tag{A.5}$$

Since

$$\Delta X_1 + \Delta X_2 = 2\Delta X_{1+\frac{1}{2}} \tag{A.6}$$

then

$$k_{1+\frac{1}{2}} = \frac{2\Delta X_{1+\frac{1}{2}}}{\frac{\Delta X_1}{k_1} + \frac{\Delta X_2}{k_2}} \tag{A.7}$$

Or in a general form

$$\left(\frac{k}{\Delta X}\right)_{x\pm\frac{1}{2}} = \frac{2}{\frac{\Delta X_x}{k_x} + \frac{\Delta X_{x\pm1}}{k_{x\pm1}}} \tag{A.8}$$

Appendix B

Derivation of the MINC Method

We demonstrate MINC method by approximating the diffusivity equation as follows:

$$\frac{1}{x}\frac{\partial}{\partial x}\left(x\frac{\partial P}{\partial x}\right)+\frac{1}{y}\frac{\partial}{\partial y}\left(y\frac{\partial P}{\partial y}\right)=\frac{\partial P}{\partial t} \tag{B.1}$$

The basis of the MINC method is that the change in the thermodynamic condition is controlled by the distance from the nearest fracture. Therefor, the matrix block is divided into nested volume based on the distance from the fracture (Wu and Pruess, 1986). Each matrix block is divided into four subregions separating by noflow boundaries as shown in Fig. B.1. Regions 1 and 3 are symmetric as well as regions 2 and 4. In regions 1 and 3, the y-component of the pressure derivative is assumed equal to zero; in regions 2 and 4, the x-component of the pressure derivative is assumed equal to zero. Because of the symmetry, only only regions 3 and 4 are considered here. The discrete form of Eq. B.1 in the x direction can be written as

$$\begin{aligned}\frac{1}{x}\frac{\partial}{\partial x}\left(x\frac{\partial P}{\partial x}\right) &= \frac{1}{x_i}\frac{\left(x\frac{\partial P}{\partial x}\right)_{i+\frac{1}{2}} - \left(x\frac{\partial P}{\partial x}\right)_{i-\frac{1}{2}}}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \\ &= \frac{1}{x_i\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right)}\left[x_{i+\frac{1}{2}}\left(\frac{P_{i+1}-P_i}{x_{i+1}-x_i}\right) - x_{i-\frac{1}{2}}\left(\frac{P_i - P_{i-1}}{x_i - x_{i-1}}\right)\right]\end{aligned} \tag{B.2}$$

Note that x is the distance from the vertex of the triangular shape of region 3, and y is the distance from the vertex of the extended triangular shape of region 4 as shown in Fig. B.2.

Note that

$$\frac{x_{i+\frac{1}{2}}}{x_i} = \frac{y_{i+\frac{1}{2}}}{y_i} \tag{B.3}$$

and

$$A_{bi} = \frac{y_{i+\frac{1}{2}} + y_{i-\frac{1}{2}}}{2}\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right) = y_i\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right) \tag{B.4}$$

Then

$$\frac{x_{i+\frac{1}{2}}}{x_i\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right)} = \frac{y_{i+\frac{1}{2}}}{y_i\left(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}\right)} = \frac{y_{i+\frac{1}{2}}}{A_{bi}} \tag{B.5}$$

where A_{bi} is the area of slice i (Fig. B.2). Substituting Eqs. B.2 and B.5 into Eq. B.1 yields

$$\frac{1}{x}\frac{\partial}{\partial x}\left(x\frac{\partial P}{\partial x}\right) = \frac{y_{i+\frac{1}{2}}}{A_{bi}}\left(\frac{P_{i+1}-P_i}{x_{i+1}-x_i}\right) - \frac{y_{i-\frac{1}{2}}}{A_{bi}}\left(\frac{P_i - P_{i-1}}{x_i - x_{i-1}}\right) \tag{B.6}$$

Using the same procedure for the y-component of Eq. B.1 yields

$$\frac{1}{y}\frac{\partial}{\partial y}\left(y\frac{\partial P}{\partial y}\right) = \frac{x_{j+\frac{1}{2}}}{A_{bj}}\left(\frac{P_{j+1}-P_j}{y_{j+1}-x_j}\right) - \frac{x_{j-\frac{1}{2}}}{A_{bj}}\left(\frac{P_j-P_{j-1}}{x_j-x_{j-1}}\right) \tag{B.7}$$

Multiplying Eqs. B.6 and B.7 by 2 (for two regions in each direction) and summing the equations yields

$$\begin{aligned} \frac{1}{x}\frac{\partial}{\partial x}\left(x\frac{\partial P}{\partial x}\right) + \frac{1}{y}\frac{\partial}{\partial y}\left(y\frac{\partial P}{\partial y}\right) = & \\ & \frac{2}{A_{b_{ring}}}\left[y_{i+\frac{1}{2}j}\left(\frac{P_{i+1j}-P_{ij}}{x_{i+1j}-x_{ij}}\right) - y_{i-\frac{1}{2}j}\left(\frac{P_{ij}-P_{i-1j}}{x_{ij}-x_{i-1j}}\right)\right] \\ & \frac{2}{A_{b_{ring}}}\left[x_{ij+\frac{1}{2}}\left(\frac{P_{ij+1}-P_{ij}}{y_{ij+1}-x_{ij}}\right) - x_{ij-\frac{1}{2}}\left(\frac{P_{ij}-P_{ij-1}}{y_{ij}-y_{ij-1}}\right)\right] \end{aligned} \tag{B.8}$$

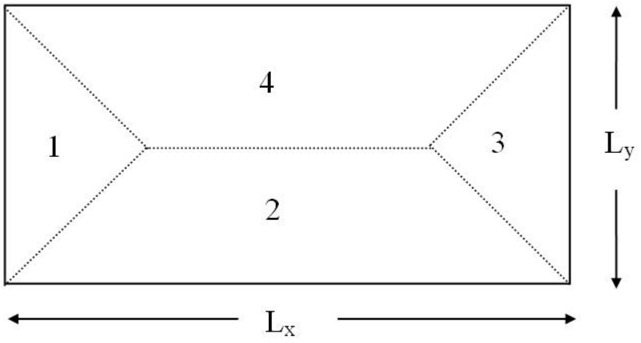


Figure B.1: MINC subregions in a matrix block

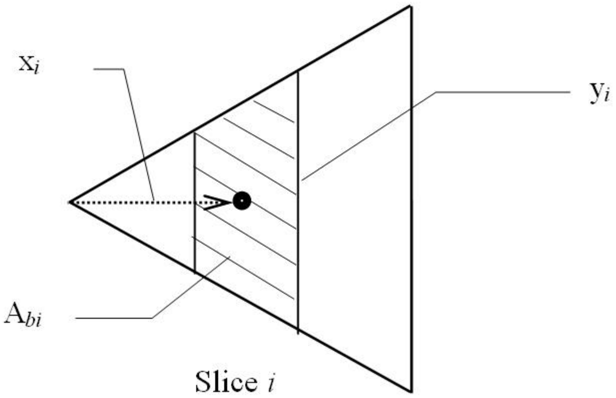


Figure B.2: Illustration of the nomenclature in subregion 3

Appendix C

Multi-Phase Multi-Component Full Tensor Formulation

The mass balance equation for component i can be written in a Cartesian system as

$$\begin{aligned}\frac{\partial(\Phi N_i)}{\partial t} &- \frac{\partial}{\partial x}\left[\sum_{j=1}^{n_p}\xi_j x_{ij}\lambda_j\left(k_{11}\frac{\partial\Phi_j}{\partial x}+k_{12}\frac{\partial\Phi_j}{\partial y}+k_{13}\frac{\partial\Phi_j}{\partial z}\right)\right]\\ &- \frac{\partial}{\partial y}\left[\sum_{j=1}^{n_p}\xi_j x_{ij}\lambda_j\left(k_{21}\frac{\partial\Phi_j}{\partial x}+k_{22}\frac{\partial\Phi_j}{\partial y}+k_{23}\frac{\partial\Phi_j}{\partial z}\right)\right]\\ &- \frac{\partial}{\partial z}\left[\sum_{j=1}^{n_p}\xi_j x_{ij}\lambda_j\left(k_{31}\frac{\partial\Phi_j}{\partial x}+k_{32}\frac{\partial\Phi_j}{\partial y}+k_{33}\frac{\partial\Phi_j}{\partial z}\right)\right]\\ &- q_i + \tau_{mfi} = 0\end{aligned} \tag{C.1}$$

The method here is defined by mapping the original grid system to an orthogonal grid system which can be considered as the computational grid system. Hence, the existing code with modified transmissibilities can be used to simulate the problem. Fig. 6.2 shows a physical and computational domain for a 2D case.

Let (x, y, z) be a right-hand Cartesian coordinate system and (ξ, η, γ) be a curvilinear coordinate system. Assume there is a one-to-one mapping between the position vector $r = [x, y, z]^T$ and $\varrho = [\xi, \eta, \gamma]^T$ then

$$\xi = \xi(x, y, z); \quad \eta = \eta(x, y, z); \quad \gamma = \gamma(x, y, z) \tag{C.2}$$

Using the above definitiones, Eq. C.1 and its counterparts can be solved in a rectangular domain that is usually called computational domain (Thompson *et al.*, 1985; Marcondes *et al.*, 2005). We define some of the terms in Eq. C.1 as

$$\begin{aligned}
E &= \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \left(k_{11}\frac{\partial \Phi_j}{\partial x} + k_{12}\frac{\partial \Phi_j}{\partial y} + k_{13}\frac{\partial \Phi_j}{\partial z} \right) \\
&= \Gamma_{xij} \left(k_{11}\frac{\partial \Phi_j}{\partial x} + k_{12}\frac{\partial \Phi_j}{\partial y} + k_{13}\frac{\partial \Phi_j}{\partial z} \right) \\
F &= \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \left(k_{21}\frac{\partial \Phi_j}{\partial x} + k_{22}\frac{\partial \Phi_j}{\partial y} + k_{23}\frac{\partial \Phi_j}{\partial z} \right) \\
&= \Gamma_{yij} \left(k_{21}\frac{\partial \Phi_j}{\partial x} + k_{22}\frac{\partial \Phi_j}{\partial y} + k_{23}\frac{\partial \Phi_j}{\partial z} \right) \\
G &= \sum_{j=1}^{n_p} \xi_j x_{ij} \lambda_j \left(k_{31}\frac{\partial \Phi_j}{\partial x} + k_{32}\frac{\partial \Phi_j}{\partial y} + k_{33}\frac{\partial \Phi_j}{\partial z} \right) \\
&= \Gamma_{zij} \left(k_{31}\frac{\partial \Phi_j}{\partial x} + k_{32}\frac{\partial \Phi_j}{\partial y} + k_{33}\frac{\partial \Phi_j}{\partial z} \right)
\end{aligned} \tag{C.3}$$

We need to transform everything from (x, y, z) coordinates to (ξ, η, ω) coordinate. Applying the chain rule, the gradient of E, F, and G in x, y, and z directions are given as follows:

$$\begin{aligned}
\frac{\partial E}{\partial x} &= \frac{\partial E}{\partial \xi}\frac{d\xi}{dx} + \frac{\partial E}{\partial \eta}\frac{d\eta}{dx} + \frac{\partial E}{\partial \gamma}\frac{d\gamma}{dx} \\
\frac{\partial F}{\partial y} &= \frac{\partial F}{\partial \xi}\frac{d\xi}{dy} + \frac{\partial F}{\partial \eta}\frac{d\eta}{dy} + \frac{\partial F}{\partial \gamma}\frac{d\gamma}{dy} \\
\frac{\partial G}{\partial z} &= \frac{\partial G}{\partial \xi}\frac{d\xi}{dz} + \frac{\partial G}{\partial \eta}\frac{d\eta}{dz} + \frac{\partial G}{\partial \gamma}\frac{d\gamma}{dz}
\end{aligned} \tag{C.4}$$

The Jacobian matrix of transformation $\mathbf{J}$ is defined as

$$\mathbf{J} = \frac{dr}{d\varrho} \tag{C.5}$$

Dividing each term of Eq. 6.9 by $\mathbf{J}$ and replacing the results in Eq. 6.7 and adding and subtracting terms of type $E\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{\mathbf{J}}\right), F\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{\mathbf{J}}\right), \ldots$ will give

$$\begin{aligned}
\frac{\partial}{\partial t}\left(\frac{\Phi N_i}{\mathbf{J}}\right) - \frac{\partial}{\partial \xi}\left(\frac{\xi_x E + \xi_y F + \xi_z G}{\mathbf{J}}\right) &- \frac{\partial}{\partial \eta}\left(\frac{\eta_x E + \eta_y F + \eta_z G}{\mathbf{J}}\right) \\
&- \frac{\partial}{\partial \gamma}\left(\frac{\gamma_x E + \gamma_y F + \gamma_z G}{\mathbf{J}}\right) \\
&- E\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_x}{\mathbf{J}}\right) + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_x}{\mathbf{J}}\right)\right] \\
&- F\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_y}{\mathbf{J}}\right) + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_z}{\mathbf{J}}\right)\right] \\
&- G\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_z}{\mathbf{J}}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_z}{\mathbf{J}}\right) + \frac{\partial}{\partial \gamma}\left(\frac{\gamma_z}{\mathbf{J}}\right)\right] \\
&- \frac{q_i}{\mathbf{J}} - \frac{\tau_{mfi}}{\mathbf{J}}
\end{aligned} \tag{C.6}$$

where $\xi_x = \frac{d\xi}{dx}, \xi_y = \frac{d\xi}{dy}, \ldots.$, and so on. It can be shown that each term in the brackets is equal to zero. Applying the chain rule to E, F, and G and inserting the results into Eq. C.6 yields

$$
\begin{aligned}
&\frac{\partial}{\partial t}\left(\frac{\Phi N_i}{\mathbf{J}}\right) \\
&-\frac{\partial}{\partial \xi}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\xi_x^2\Gamma_{xij}+\xi_y^2\Gamma_{yij}+\xi_z^2\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \xi}\right] \\
&-\frac{\partial}{\partial \eta}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\eta_x^2\Gamma_{xij}+\eta_y^2\Gamma_{yij}+\eta_z^2\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \eta}\right] \\
&-\frac{\partial}{\partial \gamma}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\gamma_x^2\Gamma_{xij}+\gamma_y^2\Gamma_{yij}+\gamma_z^2\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \gamma}\right] \\
&-\frac{\partial}{\partial \xi}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\xi_x\eta_x\Gamma_{xij}+\xi_y\eta_y\Gamma_{yij}+\xi_z\eta_z\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \eta}\right] \\
&-\frac{\partial}{\partial \xi}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\xi_x\gamma_x\Gamma_{xij}+\xi_y\gamma_y\Gamma_{yij}+\xi_z\gamma_z\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \gamma}\right] \\
&-\frac{\partial}{\partial \eta}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\xi_x\eta_x\Gamma_{xij}+\xi_y\eta_y\Gamma_{yij}+\xi_z\eta_z\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \xi}\right] \\
&-\frac{\partial}{\partial \eta}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\eta_x\gamma_x\Gamma_{xij}+\eta_y\gamma_y\Gamma_{yij}+\eta_z\gamma_z\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \gamma}\right] \\
&-\frac{\partial}{\partial \gamma}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\xi_x\gamma_g\Gamma_{xij}+\xi_y\gamma_y\Gamma_{yij}+\xi_z\gamma_z\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \xi}\right] \\
&-\frac{\partial}{\partial \gamma}\left[\sum_{j=1}^{n_p}\frac{1}{\mathbf{J}}\left(\eta_x\gamma_x\Gamma_{xij}+\eta_y\gamma_y\Gamma_{yij}+\eta_z\gamma_z\Gamma_{zij}\right)\frac{\partial \Phi_j}{\partial \eta}\right]
\end{aligned}
\tag{C.7}
$$

where the derivatives and Jacobian of the transformation can be evaluated as

$$
\begin{aligned}
\frac{\xi_x^2}{\mathbf{J}} &= \mathbf{J}\left(y_\eta z_\eta - y_\gamma z_\eta\right)^2 \\
\frac{\xi_y^2}{\mathbf{J}} &= \mathbf{J}\left(x_\eta z_\gamma - x_\gamma z_\eta\right)^2 \\
\frac{\xi_z^2}{\mathbf{J}} &= \mathbf{J}\left(x_\eta y_\gamma - x_\gamma y_\eta\right)^2 \\
\frac{\xi_x \eta_x}{\mathbf{J}} &= -\mathbf{J}\left(y_\eta z_\gamma - y_\gamma z_\eta\right)\left(y_\xi z_\gamma - y_\gamma z_\xi\right) \\
\frac{\xi_y \eta_y}{\mathbf{J}} &= -\mathbf{J}\left(x_\eta z_\gamma - x_\gamma z_\eta\right)\left(x_\xi z_\gamma - x_\gamma z_\xi\right) \\
\frac{\xi_z \eta_z}{\mathbf{J}} &= -\mathbf{J}\left(x_\eta y_\gamma - x_\gamma y_\eta\right)\left(x_\xi y_\gamma - x_\gamma y_\xi\right) \\
\frac{\eta_x^2}{\mathbf{J}} &= \mathbf{J}\left(y_\xi z_\gamma - y_\gamma z_\xi\right)^2 \\
\frac{\eta_y^2}{\mathbf{J}} &= \mathbf{J}\left(x_\xi z_\gamma - x_\gamma z_\xi\right)^2 \\
\frac{\eta_z^2}{\mathbf{J}} &= \mathbf{J}\left(x_\xi y_\gamma - x_\gamma y_\xi\right)^2 \\
\frac{\xi_x \gamma_x}{\mathbf{J}} &= -\mathbf{J}\left(y_\eta z_\gamma - y_\gamma z_\eta\right)\left(y_\xi z_\eta - y_\eta z_\xi\right) \\
\frac{\xi_y \gamma_y}{\mathbf{J}} &= -\mathbf{J}\left(x_\eta z_\gamma - x_\gamma z_\eta\right)\left(x_\xi z_\eta - x_\eta z_\xi\right) \\
\frac{\xi_z \gamma_z}{\mathbf{J}} &= -\mathbf{J}\left(x_\eta y_\gamma - x_\gamma y_\eta\right)\left(x_\xi y_\eta - x_\eta y_\xi\right) \\
\frac{\gamma_x^2}{\mathbf{J}} &= \mathbf{J}\left(y_\xi z_\eta - y_\eta z_\xi\right)^2 \\
\frac{\gamma_y^2}{\mathbf{J}} &= \mathbf{J}\left(x_\xi z_\eta - x_\eta z_\xi\right)^2 \\
\frac{\gamma_z^2}{\mathbf{J}} &= \mathbf{J}\left(x_\xi y_\eta - x_\eta y_\xi\right)^2 \\
\frac{\eta_x \gamma_x}{\mathbf{J}} &= -\mathbf{J}\left(y_\xi z_\gamma - y_\gamma z_\xi\right)\left(y_\xi z_\eta - y_\eta z_\xi\right) \\
\frac{\eta_y \gamma_y}{\mathbf{J}} &= -\mathbf{J}\left(x_\xi z_\gamma - x_\gamma z_\xi\right)\left(x_\xi z_\eta - x_\eta z_\xi\right) \\
\frac{\eta_z \gamma_z}{\mathbf{J}} &= -\mathbf{J}\left(x_\xi y_\gamma - x_\gamma y_\xi\right)\left(x_\xi y_\eta - x_\eta y_\xi\right)
\end{aligned}
\tag{C.8}
$$

and

$$\mathbf{J} = \left[x_\xi \left(y_\eta z_\gamma - y_\gamma z_\eta\right) - x_\eta \left(y_\xi z_\gamma - y_\gamma z_\xi\right) + x_\gamma \left(y_\xi z_\eta - y_\eta z_\xi\right)\right]^{-1} \tag{C.9}$$

After the mesh generation, the expression in Eqs. C.8 and C.9 can be easily evaluated using the finite difference method. For example, for a 2D case shown in Fig. C.1, $x_{\xi e}, y_{\xi_e}, x_{\eta e}$, and $y_{\eta y}$ are given by

$$\begin{aligned}
\left(\frac{\partial x}{\partial \xi}\right)_e &= \frac{x_D - x_C}{\Delta \xi} \\
\left(\frac{\partial y}{\partial \xi}\right)_e &= \frac{y_D - y_C}{\Delta \xi} \\
\left(\frac{\partial x}{\partial \eta}\right)_e &= \frac{x_A - x_B}{\Delta \eta} \\
\left(\frac{\partial y}{\partial \eta}\right)_e &= \frac{y_A - y_B}{\Delta \eta}
\end{aligned} \tag{C.10}$$

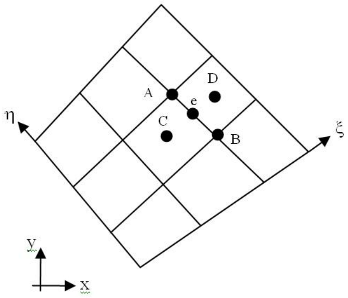


Figure C.1: Metrices evaluation

The coordinates at points C and D are obtained by interpolating the coordinates at the four neighbor corners. Whenever coordinates are not available at a specific position, for example points C and D in Fig. C.1, an interpolation procedure should be done, instead of using an interpolation in metrices (Marcondes *et al.*, 2005). A similar procedure is employed to calculate the metrices in other positions of the domain.

Integrating Eq. C.7 in time and space (control volume of Fig. 6.3) and considering a fully implicit formulation for all terms, except the derivatives of the cross terms, and dividing each term by Δt, the following equation is obtained

$$
\begin{aligned}
&\left.\frac{\phi N_i}{\mathbf{J}}\right|_p^{n+1} \frac{\Delta V}{\Delta t} - \left.\frac{\phi N_i}{\mathbf{J}}\right|_p^{n} \frac{\Delta V}{\Delta t} \\
&- \sum_{j=1}^{n_p} D_{11ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \xi}\right|_e^{n+1} \Delta\eta\Delta\gamma + \sum_{j=1}^{n_p} D_{11ij}^{n} \left.\frac{\partial \Phi_j}{\partial \xi}\right|_w^{n+1} \Delta\eta\Delta\gamma \\
&- \sum_{j=1}^{n_p} D_{12ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \eta}\right|_e^{n} \Delta\eta\Delta\gamma + \sum_{j=1}^{n_p} D_{12ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \eta}\right|_w^{n} \Delta\eta\Delta\gamma \\
&- \sum_{j=1}^{n_p} D_{13ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \gamma}\right|_e^{n} \Delta\eta\Delta\gamma + \sum_{j=1}^{n_p} D_{13ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \gamma}\right|_w^{n} \Delta\eta\Delta\gamma \\
&- \sum_{j=1}^{n_p} D_{21ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \xi}\right|_n^{n} \Delta\xi\Delta\gamma + \sum_{j=1}^{n_p} D_{21ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \xi}\right|_s^{n} \Delta\xi\Delta\gamma \\
&- \sum_{j=1}^{n_p} D_{22ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \eta}\right|_n^{n+1} \Delta\xi\Delta\gamma + \sum_{j=1}^{n_p} D_{22ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \eta}\right|_s^{n+1} \Delta\xi\Delta\gamma \\
&- \sum_{j=1}^{n_p} D_{23ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \gamma}\right|_n^{n} \Delta\xi\Delta\gamma + \sum_{j=1}^{n_p} D_{23ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \gamma}\right|_s^{n} \Delta\xi\Delta\gamma \\
&- \sum_{j=1}^{n_p} D_{31ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \xi}\right|_f^{n} \Delta\xi\Delta\eta + \sum_{j=1}^{n_p} D_{31ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \xi}\right|_f^{n} \Delta\xi\Delta\eta \\
&- \sum_{j=1}^{n_p} D_{32ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \eta}\right|_f^{n} \Delta\xi\Delta\eta + \sum_{j=1}^{n_p} D_{32ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \eta}\right|_f^{n} \Delta\xi\Delta\eta \\
&- \sum_{j=1}^{n_p} D_{33ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \gamma}\right|_f^{n+1} \Delta\xi\Delta\eta + \sum_{j=1}^{n_p} D_{33ij}^{n+1} \left.\frac{\partial \Phi_j}{\partial \gamma}\right|_f^{n+1} \Delta\xi\Delta\eta \\
&- \frac{q_{i,p}^{n+1}\Delta V}{J_p} = 0
\end{aligned}
\tag{C.11}
$$

where the tensor D contains geometric information and fluid properties. Each component of this tensor is given by

$$\begin{aligned}
D_{11ij} &= \frac{1}{\mathbf{J}}\left(\xi_x^2\Gamma_{xij} + \xi_y^2\Gamma_{yij} + \xi_z^2\Gamma_{zij}\right)\\
D_{22ij} &= \frac{1}{\mathbf{J}}\left(\eta_x^2\Gamma_{xij} + \eta_y^2\Gamma_{yij} + \eta_z^2\Gamma_{zij}\right)\\
D_{33ij} &= \frac{1}{\mathbf{J}}\left(\gamma_x^2\Gamma_{xij} + \gamma_y^2\Gamma_{xij} + \gamma_z^2\Gamma_{xij}\right)\\
D_{12ij} &= D_{21ij} = \frac{1}{\mathbf{J}}\left(\xi_x\eta_x\Gamma_{xij} + \xi_y\eta_y\Gamma_{yij} + \xi_z\eta_z\Gamma_{zij}\right)\\
D_{13ij} &= D_{31ij} = \frac{1}{\mathbf{J}}\left(\xi_x\gamma_x\Gamma_{xij} + \xi_y\gamma_y\Gamma_{yij} + \xi_z\gamma_z\Gamma_{zij}\right)\\
D_{23ij} &= D_{32ij} = \frac{1}{\mathbf{J}}\left(\eta_x\gamma_x\Gamma_{xij} + \eta_y\gamma_y\Gamma_{yij} + \eta_z\gamma_z\Gamma_{zij}\right)
\end{aligned} \tag{C.12}$$

Note that all physical properties at the each interface are evaluated at time level $n+1$, while the cross terms are evaluated at time level n. This reduces the convergence rate when the mesh is highly none-orthogonal. However, it produces the same Jacobian matrix structure as Cartesian or orthogonal corner point grids, but it is more accurate because it considers much more irregular geometry. The derivatives in Eq. C.11 are evaluating by a central differencing scheme (see Fig. 6.3 and Eqs. 6.25 and 6.26).

Appendix D

Permeability Tensor Transformation

A diagonal permeability tensor can be transformed to a full permeability tensor and vice versa. Here, we present a general formula for 3D permeability tensor transformations.

D.1 Mathematical Background

Two $n \times n$ matrices A and B are said to be similar if there exists an invertible $n \times n$ matrix P such that

$$B = P^{-1}AP \tag{D.1}$$

The term similarity transformation can be explained as the transformation of matrix A into a matrix B. Similar matrices share several properties such as rank, determinant, eigenvalues, and characteristic polynomial.

If matrix P is orthogonal, which means that $P^{-1} = P^{T}$, then the orthogonal transformation of A is expressed as

$$B = P^{T} A P \tag{D.2}$$

If A is symmetric, B will be symmetric too.

D.2 Tensor Transformation

As discussed earlier, the eigenvalues of a symmetric matrix is preserved after an orthogonal transformation. The eigenvalues of a permeability tensor is called its principle permeabilities (Bear, 1972).

Let assume $\bar{\bar{k}}$ is the full permeability tensor under the original coordinate system, and $\bar{\bar{k}}'$is the permeability tensor under the new coordinate system. The new coordinate system is derived from the original coordinate system by rotating the original coordinate system θ degree around one of its axis. Using the orthogonal tensor transformation we have

$$\bar{\bar{k}}' = U_p^T \bar{\bar{k}} U_p \tag{D.3}$$

where U_p is the rotation matrix and subscript p denotes axes of the rotation. $\bar{\bar{k}}$ and $\bar{\bar{k}}'$ are

$$\bar{\bar{k}} = \begin{bmatrix} k_{xx} & k_{xy} & k_{xz} \\ k_{yx} & k_{yy} & k_{yz} \\ k_{zx} & k_{zy} & k_{zz} \end{bmatrix} \tag{D.4}$$

$$\bar{\bar{k}}' = \begin{bmatrix} k'_{xx} & k'_{xy} & k'_{xz} \\ k'_{yx} & k'_{yy} & k'_{yz} \\ k'_{zx} & k'_{zy} & k'_{zz} \end{bmatrix} \tag{D.5}$$

There are three 3-dimensional rotation rotation matrices which correspond to rotations about x, y, and z axes. For a left-handed system, the rotation matrices are

- Rotation about x axes (Rotation in y-z plane):

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin\theta & \cos(\theta) \end{bmatrix} \tag{D.6}$$

- Rotation about y axes (Rotation in x-z plane):

$$R_{y,\theta} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{D.7}$$

- Rotation about z axes (Rotation in x-y plane):

$$R_{z,\theta} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{D.8}$$

where θ is a counterclockwise angle. For a 2D system

$$R_\theta = \begin{bmatrix} \cos\theta & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{D.9}$$

Using Eq. D.3 and rotation matrices, the following formulas are obtained for the tensor transformation for a 3D problem:

- Rotation about x axes (Rotation in y-z plane):

$$\begin{aligned} k'_{xx} &= k_{xx} \\ k'_{xy} &= k_{xy}\cos(\theta) - k_{xz}\sin(\theta) \\ k'_{xz} &= k_{xy}\sin(\theta) + k_{xz}\cos(\theta) \\ k'_{yy} &= k_{yy}\cos^2(\theta) - 2k_{yz}\sin(\theta)\cos(\theta) + k_{zz}\left[1-\cos^2(\theta)\right] \\ k'_{yz} &= k_{yy}\cos(\theta)\sin(\theta) - k_{zz}\cos(\theta)\sin(\theta) - k_{yz}\left[1-\cos^2(\theta)\right] \\ k'_{zz} &= 2k_{yz}\sin(\theta)\cos(\theta) - k_{zz}\cos^2(\theta) + k_{yy}\left[1-\cos^2(\theta)\right] \end{aligned} \tag{D.10}$$

- Rotation about y axes (Rotation in x-z plane):

$$\begin{aligned} k'_{xx} &= k_{xx}\cos^2(\theta) + 2k_{xz}\sin(\theta)\cos(\theta) + k_{zz}\left[1-\cos^2(\theta)\right] \\ k'_{xy} &= k_{xy}\cos(\theta) - k_{yz}\sin(\theta) \\ k'_{xz} &= -k_{xx}\sin(\theta)\cos(\theta) + k_{zz}\cos(\theta)\sin(\theta) - k_{xz}\left[1-2\cos^2(\theta)\right] \\ k'_{yy} &= k_{yy} \\ k'_{yz} &= -k_{xy}\sin(\theta) + k_{yz}\cos(\theta) \\ k'_{zz} &= -2k_{xz}\sin(\theta)\cos(\theta) + k_{zz}\cos^2(\theta) + k_{xx}\left[1-\cos^2(\theta)\right] \end{aligned} \tag{D.11}$$

- Rotation about z axes (Rotation in x-y plane):

$$\begin{aligned}
k'_{xx} &= k_{xx}\cos^2(\theta) - 2k_{xy}\sin(\theta)\cos(\theta) - k_{yy}\cos^2(\theta) \\
k'_{xy} &= (k_{xx} - k_{yy})\cos(\theta)\sin(\theta) + 2k_{xy}\cos^2(\theta) - k_{xy} \\
k'_{xz} &= k_{xz}\cos(\theta) - k_{yz}\sin(\theta) \\
k'_{yy} &= 2k_{xy}\sin(\theta)\cos(\theta) + k_{yy}\cos^2(\theta) - k_{xx}\left[1 - \cos^2(\theta)\right] \\
k'_{yz} &= k_{xz}\sin(\theta) + k_{yz}\cos(\theta) \\
k'_{zz} &= k_{zz}
\end{aligned} \tag{D.12}$$

D.3 Principle Values of Permeability Tensor

Equations D.10 through D.12 can be used to transform a permeability tensor after the coordinate system is rotated. The diagonal form of a full permeability tensor can be determined by rotating the coordinate system three times. The three rotation angles are principle directions of the permeability tensor. The three angle, θ_{xy}, θ_{xz}, and θ_{yz} are given by

$$\tan(\theta_{xy}) = \frac{2k'_{xy}}{k'_{xx} - k'_{yy}} \tag{D.13}$$

$$\tan(\theta_{xz}) = \frac{2k'_{xz}}{k'_{xx} - k'_{zz}} \tag{D.14}$$

$$\tan(\theta_{yz}) = \frac{2k'_{yz}}{k'_{yy} - k'_{zz}} \tag{D.15}$$

Appendix E

GPAS Input Keywords

E.1 Flags

Keyword	Description
DUALPORO	Flag for the dual porosity option
OUTPUT_PERM	Printing out the pressure for each matrix subgrid
OUTPUT_SATM	Printing out the saturation for each matrix subgrid
OUTPUT_PMAVE	Printing out the average pressures for matrix blocks
OUTPUT_SATMAVE	Printing out the average saturations for matrix blocks
OUTPUT_VISM	Printing out the phase viscosities for matrix blocks
OUTPUT_NPHM	Printing out the presence of phases for matrix blocks
OUTPUT_DENM	Printing out the matrix phase
OUTPUT_RELPM	Printing out the matrix relative permeabilities
OUTPUT_IFTM	Printing out the matrix interfacial tensions
OUTPUT_TRAPM	Printing out the matrix trapping numbers
ITRAPM	Flag to calculate trapping number

E.2 Data Variables

Keyword	Description
NSH	Number of subgrids in horizontal direction
NSV	Number of subgrids in vertical direction
DXM()	Matrix block size array in X diection
DYM()	Matrix block size array in Y diection
DZM()	Matrix block size array in Z diection
MCAPP	Capillary pressure option 0 ⇒ Ignore capillary pressure in the matrix (default) 1 ⇒ Include capillary pressure in the matrix
MGRAV	Gravity option 0 ⇒ Ignore gravity effect in the matrix (default) 1 ⇒ Include gravity effect in the matrix
VFVDM()	Volume fraction of subgrids in the vertical direction
ISEALT	Option for sealing the top of the matrix block 0 ⇒ Matrix subgrid top is not sealed (default) 1 ⇒ Matrix subgrid top is sealed
ISEALB	Option for sealing the bottom of the matrix block 0 ⇒ Matrix subgrid bottom is not sealed (default) 1 ⇒ Matrix subgrid bottom is sealed
ISEALS	Option for sealing sides of the matrix block 0 ⇒ Matrix subgrid sides is not sealed (default) 1 ⇒ Matrix subgrid sides is sealed
ISUBEQ	Matrix gridblock anisotropy option 0 ⇒ Permeability anisotropy is not treated (default) 1 ⇒ Equivalent matrix block permeability is used

Keyword	Description
MPOROSITY()	Array of matrix subgridding porosities
MXPERM()	Array of matrix subgridding permeabilities in X direction
MYPERM()	Array of matrix subgridding permeabilities in Y direction
MZPERM()	Array of matrix subgridding permeabilities in Z direction
MSWINI()	Array of matrix subgridding initial water saturations
MPINI()	Array of matrix subgridding initial pressures
MVISM()	Array of matrix subgridding water viscosities
ZXYM()	Array of matrix subgridding compositions
RELPM	Matrix relative permeability option 1 ⇒ Table lookup (default) 2 ⇒ Function based
NRELFUNM	The relative permeability model Valid with RELPM = 2 1 ⇒ For Corey-type exponential function
ENDPTM()	Relative permeability endpoint for matrix media
SRM()	Residual saturation for matrix media
EXPNM()	Relative permeability exponents for matrix media
PCGOM()	Oil and gas capillary pressure in matrix media
PCOWM()	Oil and water capillary pressure in matrix media
ENDPTLOWM()	Relative permeability endpoint at low trapping number
ENDPTHIGHM()	Relative permeability endpoint at high trapping number
SRLOWM()	Residual phase saturation at low trapping number
SRHIGHM()	Residual phase saturation at high trapping number
EXPNLOWM()	Relative permeability exponent at low trapping number
EXPNHIGHM()	Relative permeability exponent at high trapping number

Keyword	Description
TLM()	Capillary desaturation parameters
TAULM()	Capillary desaturation parameters

Appendix F

Input Files

F.1 1D Waterflood Case

```
TITLE(2)="1D TEST WATER FLOOD"

DESCRIPTION()=
"THICKNESS (FT) : 30 "
"LENGTH (FT)    : 80 "
"WIDTH (FT)  : 80  "
"GRID BLOCKS  : 8x1x1"

COMPOSITIONAL_MODEL
DUAL_POROSITY
$DEBUGS
$TIMEEND = 10
TIMEEND = 50.0
$ I/O OPTIONS
OUTLEVEL = 1
OUTPUT_PRE
OUTPUT_SAT
```

```
OUTPUT_VIS
OUTPUT_DEN
$OUTPUT_NPH
$OUTPUT_FED
$OUTPUT_GAS
$OUTPUT_OIL
$OUTPUT_WEL
$SPLINEOUT
$GEOMOUT
PROCOUT
$FEWMSG = FALSE

OUTPUT_HIS

$ Dual Porosity Output Options
$OUTPUT_PREM
$OUTPUT_SATM
$OUTPUT_PMAVE
$OUTPUT_SATMAVE
$OUTPUT_VISM
$OUTPUT_NPHM
$OUTPUT_DENM

$TDPVOPT
OUTPUT_TIME(1 TO 30) =  1. 5. 10. 20. 30. 40. 50.

$TIME_ENLARGE
$NO_CRASH

ISTEP(,,) = 1
JSTEP(,,) = 1
KSTEP(,,) = 1

$ DEBUGGING OPTION
$BUGKEY(15)

$ FAULT BLOCK AND MESH DATA
METHOD = 2
DOWN() = 0 0 1
MES="cart"
NX(1) = 8 NY(1) = 1  NZ(1) = 1
DX() = 80 DY() = 80 DZ() = 30

$ CORNER OF BLOCK 1,1,1
XYZ111(1 TO 3,1) = 0 0 0
```

```
$ SUBGRID DATA (Added by Reza)
NSH = 4
NSV = 4
DXM() = 80 DYM() = 80 DZM() = 30
MGRAV = 1
KBNDRY = 0
$ VFVDM()= 0.5
$ ISUB1D = 1
$ ISEALT = 1
$ ISEALB = 1
$ ISEALS = 1
$ ISHAPE = 1
$ ISUBEQ = 1

$ COMPOUND NAMES
COMPOUND(1) = "C10"


$ COMPOUND CRITICAL TEMPERATURES
CRIT() 1111.8

$ COMPOUND CRITICAL PRESSURES
CRIP()  304.0

$ COMPOUND CRITICAL VOLUMES
CRIV() 12.087
$CRIV() 10.087

$ COMPOUND ACEN
ACEN() 0.488

$ COMPOUND MOL WEIGHTS
MOLW() 142.3

$ COMPOUND PARA
PARA() 431.0

$ MAX NUMBER OF PHASES
NPHASE = 3

$ Initial rock & water properties
ROCKZ = 0.0000  ROCKP = 14.7
H2OZ = 0.000003  H2OP = 14.7     H2OD = 3.467
SURTF = 60.0  SURPS = 14.7
RESTF = 60.0
```

```
$ TOLERANCE
TOL_FLASH = 0.0001
TOL_VOLUME = 0.0001
TOL_MASS = 0.0001
TOL_WATER = 0.0001
TOL_MAT = 0.0001
MAXNEWT = 100

$ POROSITY
POROSITY1() = 0.01
MPOROSITY1() = 0.19

$ PERMEABILITIES

XPERM1() = 500
MXPERM1() = 1

YPERM1() = 500
MYPERM1() = 1

ZPERM1() = 500
ZPERMM1() = 1

$ INITIAL WATER SATURATION
SWINI1() = 0.25
MSWINI1() = 0.25

$ INITIAL WATER CELL PRESSURE
PINI1() = 4000.0
MPINI1() = 4000.0

$ INITIAL PHASE VISCOSITIES AT EACH CELL
VIS1() = 0.5
MVIS1() = 0.5

IOILVIS
OILVIS 2

$ INITIAL COMPOSITIONS
ZXY1(,,,1) = 1.000
ZXYM1() = 1.000

$ TRACER DATA
$ NUMERICAL METHOD FOR CHEMICAL FLOODING
IMPAQCOMP
```

```
NAQCOMP 1
AQCOMPNAM() = "TRACER"
AQCOMPTYPE() = 1
AQCOMPINIT() = 0.00

$ RELPERM DATA
$ RELP 1 for table lookup, 2 for function based
RELP  2
RELPM  2
$ NRELFUN 1 for corey, more to be added later
NRELFUN  1
NRELFUNM  1
$ data for each phase : water, phase 2 and phase 3
ENDPT() =  1.0 1.0 0.0
SR() = 0.001 0.001 0.0
EXPN() = 1.0 1.0 0.0

ENDPTM() =  0.2 0.92 0.0
SRM() = 0.25 0.3 0.0
EXPNM() = 1.18 1.8 0.0


PCGO(1) Block        $ GAS-OIL CAPILLARY PRESSURE - ROCK TYPE 1
 Interpolation Linear
 Extrapolation Same
 Data
 0.0   0.0,
 1.0   0.0
EndBlock

PCOW(1) Block        $ WATER-OIL CAPILLARY PRESSURE - ROCK TYPE 1
 Interpolation Linear
 Extrapolation Same
 Data
0.0010    4.000103 ,
0.0510    2.839109 ,
0.1010    1.977988 ,
0.1510    1.349804 ,
0.2010    0.899974 ,
0.2510    0.584513 ,
0.3010    0.368456 ,
0.3510    0.224427 ,
0.4010    0.131361 ,
0.4510    0.073373 ,
0.5010    0.038761 ,
0.5510    0.019138 ,
```

```
0.6010    0.008691 ,
0.6510    0.003550 ,
0.7010    0.001262 ,
0.7510    0.000371 ,
0.8010    0.000083 ,
0.8510    0.000012 ,
0.9010    0.000001 ,
0.9510    0.000000
EndBlock

PCGOM(1) Block        $ GAS-OIL CAPILLARY PRESSURE FOR ROCK MATRIX
  Interpolation Linear
  Extrapolation Same
  Data  0.0  0.0 , 1. 0.0
EndBlock

PCOWM(1) Block        $ WATER-OIL CAPILLARY PRESSURE FOR ROCK MATRIX
  Interpolation Linear
  Extrapolation Same
  Data
0.2500    5.7893 ,
0.2750    5.1639 ,
0.3000    4.5743 ,
0.3250    4.0203 ,
0.3500    3.5022 ,
0.3750    3.0197 ,
0.4000    2.5730 ,
0.4250    2.1621 ,
0.4500    1.7868 ,
0.4750    1.4473 ,
0.5000    1.1436 ,
0.5250    0.8755 ,
0.5500    0.6433 ,
0.5750    0.4467 ,
0.6000    0.2859 ,
0.6250    0.1608 ,
0.6500    0.0715 ,
0.6750    0.0179 ,
0.7000    0.0000
EndBlock

NUMWELL=2

WELLNAME(1) = "INJECTOR"
KINDWELL(1) = 2
PRLIMIT(1) = 10000
```

```
WELLTOP(1 TO 3,1,1) =    40. 40. 0.
WELLBOTTOM(1 TO 3,1,1) = 40. 40. 30.
DIAMETER(1,1) = 1.
WELLPQ(1) Block
  Interpolation Linear
  Extrapolation Constant
   Data  0.  498.7
EndBlock

WELLNAME(2) = "PRODUCER"
KINDWELL(2) = 3
WELLTOP(1 TO 3,1,2) =    600.  40.  0.
WELLBOTTOM(1 TO 3,1,2) = 600.  40.  30.
DIAMETER(1,2) = 1.
WELLPQ(2) Block
  Interpolation Linear
  Extrapolation Constant
   Data  0.  3900.
EndBlock

EndInitial

$ TRANSIENT DATA INPUT BLOCKS

BeginTime    0.0
$ INJECTED FLUID COMPOSITION
WZ() 0.000 1.000
TIME_CONTROL = 2
DELTIM = 0.00008 DTIMMUL = 1.0  DTIMMAX = 0.1 DTIMMIN = 0.00008
DPMAX = 1000. DSMAX = 1000. DCMAX = 1000. DAQCMAX = 1000.
EndTime
```

Nomenclature

Roman Symbols

a_{31}	Adsorption model parameters for surfactant	
a_{32}	Adsorption model parameters for surfactant	
a_{41}	Adsorption model parameters for polymer	
a_{42}	Adsorption model parameters for polymer	
A_{p1}	Viscosity model parameters for polymer	
A_{p2}	Viscosity model parameters for polymer	
A_{p3}	Viscosity model parameters for polymer	
B	Formation volume factor	$\frac{STB}{bbl}$
b_3	Adsorption model parameters for surfactant	
b_4	Adsorption model parameters for polymer	
c_f	Fracture compressibility	$\frac{1}{psia}$
C_i	Volume of i^{th} component in fluid per fluid volume	fraction
C_i^f	Volume of i^{th} component in fluid per pore volume	fraction
c_m	Matrix compressibility	$\frac{1}{psia}$
C_{3max}^s	Height of binodal curve at salinity s	
C_{SEL}	Lower effective salinity	$\frac{meq}{ml}$

C_{SEOP}	Optimal effective salinity	$\frac{meq}{ml}$
C_{SEU}	Upper effective salinity	$\frac{meq}{ml}$
C_{SE}	Effective salinity	$\frac{meq}{ml}$
C_{ij}	Volume of i^{th} component in j^{th} phase per volume of j^{th} phase	fraction
D	Depth	ft
g	Gravitational acceleration	$\frac{m}{s^2}$
g_c	Gravitational conversion factor	$\frac{lb_m-ft}{lb_f-s^2}$
h_{sub}	Matrix subgrid thickness	ft
K_f	Fracture permeability	
K_i	The equilibrium ratio of component i	
K_r	Relative permeability	
K_t	Total permeability	
L_j	Ratio of moles in phase j to the total number of moles on the mixture	
L_x	Matrix subgrid length in the x direction	ft
L_y	Matrix subgrid length in the y direction	ft
M_i	Molecular weight of i^{th} component	$\frac{lb}{lbmol}$
Mw	Molecular weight	
n_a	Total number of aqueous components	
n_c	Total number of hydrocarbon components	
N_i	Flux vector of species i	
N_i	Mole of i^{th} component per pore volume	$\frac{lbmol}{ft3}$
N_i^f	Mole of i^{th} component in fluid per pore volume	$\frac{lbmol}{ft3}$

n_j	Relative permeability exponent of j^{th} phase	
n_p	Total number of phases	
$N_{oil,3}$	Mole of hydrocarbons in microemulsion phase per pore volume	$\frac{lbmol}{ft3}$
N_{Tl}	Trapping number of l^{th} phase	fraction
$N_{T,3}$	Mole of all the components in microemulsion phase per pore volume	$\frac{lbmol}{ft3}$
P^0	Reference pressure	$psia$
P_C	Critical pressure	
P_f	Fracture pressure	psia
P_m	Matrix pressure	psia
R^v	Residual of volume constraint	
R^w	Residual of water material balance equation	$\frac{lbmol}{day}$
R_i	Mass rate of production of species i	
R_i^f	Residual of fugacity equation of i^{th} component	
R_i^m	Residual of material balance equation of i^{th} component	$\frac{lbmol}{day}$
r_o	Equivalent well gridblock radius	ft
r_w	Well radius	ft
S_j	Saturation of phase j	fraction
S_p	Viscosity model parameters for polymer	
S_{jr}	Residual saturation of j^{th} phase	fraction
T_C	Critical temperature	
T_r	Reduced temperature	
V_b	Bulk volume for a cell	ft^3

V_p	Pore volume for a cell	ft^3
W_i	Overall concentration of species i	
x_{ij}	Mole fraction of component i in phase j	fraction
Z	Compressibility factor	
Z_i	The overall mole fraction of component i	

Greek Symbols

$\alpha_1, \ldots, \alpha_5$	Microemulsion viscosity correlation parameters	
$\bar{\nu}_j$	Specific molar volume of phase j	$\frac{ft^3}{lbmol}$
γ_j	Specific gravity of phase j	$\frac{psia}{ft}$
λ_j	Relative mobility of phase j	$\frac{1}{cp}$
μ	Viscosity	cp
μ_i	The chemical potential of component i	
Φ	Flow potential	psia
ϕ	Porosity	fraction
ϕ_f	Fracture porosity	fraction
ϕ_m	Matrix porosity	fraction
ϕ_r	Porosity at a reference pressure P^0	
ϕ_{ij}	Fugacity coefficient of componet i in phase j	
ρ	Density	$\frac{lb_m}{ft^3}$
$\sigma_{ll'}$	IFT between displaced (l) and displacing (l') phases	
σ	Shape factor	
σ_{23}	Interfacial tension between oleic and microemulsion phases	$\frac{dyne}{cm}$

σ_{ow}	Interfacial tension between oleic and aqueous phases	$\frac{dyne}{cm}$
τ_j	Trapping number model parameter of j^{th} phase	
τ_{m-f}	Matrix-fracture transfer function flow	$\frac{ft^2}{s}$
ξ_j	Molar density of phase j	$\frac{lbmol}{ft^3}$

Superscripts

High	High trapping number
Low	Low trapping number

Subscripts

α	Phase
f	Fracture
g	Gas
m	Matrix
o	Oil
w	Water

Other Symbols

$\bar{\bar{k}}$	Permeability tensor	
Δx	Gridblock length in the x direction	
Δy	Gridblock length in the y direction	
Δz	Gridblock length in the z direction	
$\hat{C}_i$	Adsorbed volume of i^{th} component per pore volume	fraction
$\tilde{C}_i$	Overall volume of i^{th} component per pore volume	fraction
$\vec{\vec{K}}_{ij}$	Dispersion coefficient tensor of component i in phase j	$\frac{ft^2}{d}$

$\vec{R}_i$	Residual vector for i^{th} gridblock
$\vec{u}_j$	Superficial velocity of phase j
N_b	The number of matrix subgrids
NM	Total number of matrix blocks in a fracture gridblock

Acronyms

CMC	Critical Micelle Concentrartion
GPAS	General Purpose Adaptive Simulator
IPARS	Integrated Parallel Accurate Reservoir Simulation
MPFA	MultiPoint Flux Approximation
MPI	Message Passing Interface
PETSc	The Portable, Extensible Toolkit for Scientific Computations
UTCHEM	University of Texas Chemical Simulator

Bibliography

Aavastsmark, I., T. Barkvo, and T. Manneseth: "Control Volume Discretization Methods for 3D Quadrilateral Grids in Inhomogeneous, Anisotropic Reservoirs," *Soc. Pet. Eng. J.*, pp. 146-154, June 1998.

Abate, J., P. Wang, and K. Sepehrnoori: "Parallel Compositional Reservoir Simulation on a Cluster of PCs," presented at the Fifth SIAM Conference on Mathematical and Computational Issues in the Geosciences, San Antonio, TX, March 22-24, 1999.

Aldejain, A.A.: "Implementation of Dual Porosity Model in a Chemical Flooding Simulator," PhD. dissertation, The University of Texas at Austin, TX, 1999.

Al-Shaalan, T.M., L.S.K. Fung, and A.H. Dogru: "A Scalable Massively Parallel Dual-Porosity Dual-Permeability Simulator for Fractured Reservoirs with Super-K Permeability," Paper SPE presented at SPE Annual Technical Conference and Exhibition, Denver, Colorado, October 5-8, 2003.

Arbogast, T., P.T. Keenan, M.F. Wheeler, and I. Yotov: "Logically Rectangular Mixed Methods for Darcy Flow on General Geometry," SPE Reservoir Simulation Symposium, San Antonio, Texas, February 12-15, 1995.

Aziz, K. and A. Settari: *Petroleum Reservoir Simulation*, Applied Science Publishers, Ltd., London, 1979.

Baca, R.G., R.C. Arnett, and D.W. Langford: "Modeling fluid flow in fractured-porous rock masses by finite-element techniques," *Int. J. Num. Meth. Fluids* Vol. 4, 337, 1984.

Baker, L.E., A.C. Pierce, and K.D. Luks: "Gibbs Energy Analysis of Phase Equilibria," *Soc. Pet. Eng. J.*, vol. 22, No. 5, 731-742, 1982.

Balay, S., W. Gropp, L.C. McInnes, and B. Smith: "PETSc 2.0 User Manual," Argonne National Laboratory, ANL-95/11 - Revision 2.0.22 ,April 1998.

Baliga, B. and S. Patankar: "A New Finite-Element Formulation for Convection-Diffusion Problems," *Numer. Heat Transfer* vol. 3, 393-409, 1980.

Barenblatt, G.E., I.P. Zheltov and I.N. Kochina: "Basic Concepts in the Theory of Seepage of Homogeneous Liquids in Fissured Rocks," *J. Appl. Math. Mech.* (USSR) vol. 24, No. 5, 1960.

Bastian, P., R. Helming, H. Jakobs, and V. Reichenberger: "Numerical Simulation of Multiphase Flow in Fractured Porous Media," *Numerical Treatment of Multiphase Flows in Porous Media* , 1-18, 2000.

Beckner, B.L., A. Firoozabadi, and K. Aziz: "Modeling Transverse Imbibition in Double Porosity Simulator," Paper SPE 17414 presented at the California Regional Meeting, Long Beach, March 23-25, 1987.

Beckner, B.L., H.M. Chan, A.E. McDonald, S.O. Wooten, and T.A. Jones: "Simulating Naturally Fractured Reservoirs Using a Subdomain Method," Paper SPE 21241 presented at the Society of Petroleum Engineers Symposium on Reservoir Simulation, Anaheim, CA, Feb. 17-20, 1991.

Bear, J.: *Dynamic of Fluids in Porous Media*, American Elsevier, New York, 1972.

Camilleri, D., Fil, A., Pope, G.A., Rouse, B.A. and Sepehrnoori, K.: "Comparison of an Improved Compositional Micellar/Polymer Simulator With Laboratory Core Floods," SPE. Reservoir Engineering, November 1987.

Chang, Yih-Bor: "Development and Application of an Equation of State Compositional Simulator," Ph.D. dissertation, The University of Texas at Austin, TX, 1990.

Chen, J.: "New Approaches to Dual Porosity Modeling of Waterflooding in Naturally Fractured Reservoirs," PhD. dissertation, The University of Texas at Austin, TX, 1993.

Chen, W.H. and R.E. Fitzmorris: "A Thermal Simulator for Naturally Fractured Reservoirs." Paper SPE 16008 presented at the Society of Petroleum Engineers Symposium on Reservoir Simulation, San Antonio, February 1-4, 1987.

Chien, M.C.H., Vasserman, M.L., Yardumian, H.E., and Chung E.Y.: "The Use of Vectorization and Parallel Processing for Reservoir Simulation," Paper SPE 16025 presented at the SPE Reservoir Simulation Symposium, San Antonio, Texas, February 1-4, 1987.

Chien, M.C.H. and E.J. Northrup: "Vectorization and Parallel Processing of Local Refinement and Adaptive Schemes in a General Purpose Reservoir Simulator," Paper SPE 25258, presented at the 12th SPE Symposium on Reservoir Simulation, New Orleans, 1993.

Collins, R.E.: *Flow of Fluid Through Porous Media*, 275 pp., Reinhold, New York, 1961.

Dean, R.H. and L.L. Lo: "Development of a Naturally Fractured Reservoir Simulator and Examples of Its Use," Paper SPE 14110 presented at the Society of Petroleum Engineers International Meeting on Petroleum Engineering, Beijing, China, March 17-20, 1986.

Delshad, M.: "Trapping of Micellar Fluids in Berea Sandstone," Ph.D. dissertation, The University of Texas at Austin, TX, 1990.

Delshad, Mojdeh, G. A. Pope and K. Sepehrnoori: " A Compositional Simulator for Modeling Surfactant Enhanced Aquifer Remediation," *Journal of Contaminant Hydrology*, Vol. 23, p 303-327, 1996.

Delshad, M., M. Delshad, D. Bhuyan, G.A. Pope, and L.W. Lake: "Effect of Capillary Number of the Residual Saturation of a Three-Phase Micellar Solution," Paper SPE 14911, SPE Fifth Symposium of Enhanced Oil Recovery, Tulsa, OK, April 1986.

Dogru, A.H., K.G. Li, H.A. Sunaidi, W.A. Habiballah, L. Fung, N. Al-Zamil, D. Shin, A.E. McDonald, and N.K. Srivastava: "A Massively Parallel Reservoir Simulator for Large Scale Reservoir Simulation," Paper SPE 51886 presented at the SPE Reservoir Simulation Symposium, Houston, TX, February 14-17, 1999.

Durlofsky, L.J.: "Numerical Calculation of Equivalent Grid Block Permeability Tensors for Heterogeneous Porous Media," *Water Resources Research*, Vol. 27, No. 5, 699-708, May 1991.

Edwards, M.G.: "Symmetric Flux Continuous Positive Approximation of the Elliptic Full Tensor pressure Equation in Local Conservative Form," Paper SPE 29147 presented at the Society of Petroleum Engineers Symposium on Reservoir Simulation, San Antonio, TX, February 12-15, 1995.

Edwards, M.G.: "Unstructured, Control-Volume Distributed, Full Tensor Finite-Volume Schemes with Flow-Based Grids," *Comput. Geosci.*, *6*, p 433-452, 2002.

Fayers, F.J. and J.P. Matthews: "Evaluation of Normalized Stone's Methods for Estimating Three-Phase Relative Permeabilities," *SPE Journal*, Vol 24, 224-232, 1984.

Fjerstad, P.A., A.S. Sikandar, H. Cao, J. Liu, and W. Da Sie: "Next Generation Parallel Computing for Large-Scale Reservoir Simulation," Paper SPE 97358 presented at SPE International Improved Oil Recovery Conference in Asia Pacific, Kuala Lumpur, Malaysia, December 5-6, 2005.

Flory, P.J.: *Principles of Polymer Chemistry*, Ithaca, New York, Cornell University Press, 1953.

Fu, Y., Y.-K. Yang, and M. Deo: "Three-Dimentional, Three-Phase Discrete-Fracture Reservoir Simulator Based on Control Volume Finite Element (CVFE) Formulation," *SPE Journal*, 2005.

Geigar, S., S. Roberts, S. Matthai, and C. Zoppou: "Combining Finite Volume and Finite Element Methods to Simulate Fluid Flow in Geological Media," *ANZIAM J.*, 44(*E*), C180-C201, 2003.

Gilman, J.R. and H. Kazemi: "Improvements in Simulation of Naturally Fractured Reservoirs," *Soc. Pet. Eng. J.*, August 1983.

Gilman, J.R.: "An Efficient Finite-Difference Method for Simulating Phase Segregation in Matrix Blocks in Double Porosity Reservoirs," *Soc. Pet. Eng. Res. Eng.*, July 1986.

Golder Associates, Inc.: " FracMan User's Manual," Release 7.00

Gosset, H., G. Heyen, and B. Kalitventzeff: "An Efficient Algorithm to Solve Cubic Equations of State," *Fluid Phase Equilibria*, Vol. 25, 1986.

Han, C., M. Delshad, K. Sepehrnoori, and G.A. Pope: "A Fully Implicit, Parallel, Compositional Chemical Flooding Simulator," Paper SPE 97217 presented at SPE Annual Technical Conference and Exhibition, Dallas, Texas, October 9-12, 2005.

Hand, D.B.: "Dineric distribution: I. The distribution of a consolute liquid between two immiscible liquids," *Journal of Physics and Chemistry*, Vol. 34, 1939.

Hirasaki, G.J. and G.A. Pope: "Analysis of Factors Influencing Mobility and Adsorption in the Flow of Polymer Solution Through Porous Media," *SPE Journal*, August 1974.

Hirasaki, G.J.: "Application of the Theory of Multicomponent, Multiphase Displacement to Three-Component, Two-Phase Surfactant Flooding," *SPE Journal*, 1981.

Hoteit, H. and A. Firoozabadi: "Compositional Modeling by the Combined Discontinuous Galerkin and Mixed Methods," *SPE Journal*, Vol. 11, 19-24, March 2006.

Hoteit, H. and A. Firoozabadi: "Compositional Modeling of Discrete-Fractured Media Without Transfer Functions by the Discontinuous Galerkin and Mixed Methods," *SPE Journal*, September 2006.

Huh, C.: "Interfacial Tension and Solubilization Ability of a Microemulsion Phase that Coexists with Oil and Brine," *Journal of Colloid Interface Science*, 71, 1979.

Jin, M.: "A Study of Non-Aqueous Phase Liquid Characterization and Surfactant Remediation," PhD dissertation, University of Texas at Austin, 1995.

John, A.K.: "Development of a Hybrid Approach to Couple Chemical and Compositional Models in an Implicit Parallel Simulator," M.Sc. Thesis, The University of Texas at Austin, 2003.

Karimi-Fard, M., A. Firoozabadi: "Numerical Simulation of Water Injection in Fractured Media Using the Discrete-Fracture Model and the Galerkin Method," *Soc. Pet. Eng.*, April 2003.

Karimi-Fard, M., L.J. Durlofsky, and K. Aziz: "An Efficient Discrete-Fracture Model Applicable for General-Purpose Reservoir Simulators," *SPE Journal* Vol. 9, 227-236, 2004.

Kasap, E., and L. W. Lake : "An Analytical Method to Calculate the Effective Permeability Tensor of a Grid Block and Its Application in an Outcrop Study," Paper presented at the SPE Symposium on Reservoir Simulation, Soc of Pet. Eng., Houston, Texas, 1989.

Kazemi, H.: "Pressure Transient Analysis of Naturally Fractured Reservoirs," *SPE Journal*, p. 451-462, December 1969.

Kazemi, H. and L.S. Merrill Jr., K.L. Porterfield, and P.R. Zeman: "Numerical Simulation of Water-Oil Flow in Naturally Fractured Reservoirs," *SPE Journal*, December 1976.

Killough, J.E. and R. Bhogeswara: "Simulation of Compositional Reservoir Phenomena on a Distributed Memory Parallel Computer," *J. of Petroleum Technology*, November 1991.

Killough J.E., Camilleri, D., and Harlow B.: "A Parallel Simulator on Local Grid Refinement," Paper SPE 37978 presented at the SPE Symposium on Reservoir Simulation, Dallas, June 8-11, 1997.

Killough, J.E. and C.A. Kossack: "Fifth comparative simulation project: Evaluation of miscible flood simulations," Paper SPE 16000 presented at the Ninth SPE Symposium on Reservoir Simulation, San Antonio, TX, February 1987.

Kim, J-G. and M.D. Deo: "Finite Element Discrete-Fracture Model for Multiphase Flow in Porous Media," *AIChE J.*, Vol. 46, 1120, 2000.

Kleppe, J. and R.A. Morse: "Oil Production from Fractured Reservoirs by Water Displacement," Paper SPE 5084 presented at the SPE Annual Meeting, Houston, TX, October 6-9, 1974.

Kohar, Gill and J.E. Killough: "An Asynchronous Parallel Linear Equation Solution Technique," Paper SPE 29142 Presented at the 13th SPE Symposium on Reservoir Simulation, San Antonio, TX, Febraury 12-15, 1995.

Lake, L.W. :*Enhanced Oil Recovery, Prentice Hall* , Englewwod Cliffs, NJ, 1989.

Lee, S.H., L.J. Durlofsky, M.F. Lough, and W.H. Chen: "Finite Difference Simulation of Geologically Complex Reservoirs With Tensor Permeability," Paper SPE 3802 presented at the Society of Petroleum Engineers Reservoir Simulation Symposium, Dallas, TX, June 8-11, 1997.

Lee, S.H., H. Tchelepi, and L.F. DeChant: "Implementation of the Flux-Continuous Finite Difference Method for Stratigraphic Hexahedron Grids," Paper SPE 51901 presented at the Society of Petroleum Engineers Reservoir Simulation Symposium, Houston, TX, February 14-17, 1999.

Li, B.: "Implementation of Full Permeability Tensor Representation in a Dual Porosity Reservoir Simulator," PhD. dissertation, The University of Texas at Austin, TX, 2001.

Litvak, B.L.: "Simulation and Characterization of Naturally Fractured Reservoirs," Proceeding of the Reservoir Characterization Technical Conference, Dallas, TX, April 29 - May 1, 1985.

Liu, W., J. Cao, and A. Mezzatesta: "Parallel Reservoir Simulation on Shared and Distributed Memory System," Paper SPE 64797 presented at the SPE International Oil and Gas Conference and Exhibition in China held in Beijing, China, November 710, 2000.

Lohrenz, J., B.G. Bray, and C.R. Clark: "Calculating Viscosities of Reservoir Fluids from Their Compositions," *Trans.*, AIME, 1964.

Long, J.C.S., J.S. Remer, C.R. Wilson, and P.A. Witherspoon :"Porous Media Equivalent for Networks of Discontinuous Fractures" *Water Resources Research*, Vol. 18, No. 3, p. 645-658, June 1982.

Lough M.F., S.H. Lee, and J. Kamath: "A New Method to Calculate the Effective permeability of Grid Blocks used in the Simulation of naturally Fractured Reservoirs," Paper SPE 36730 presented at the Society of Petroleum Engineers Annual technical Conference and Exhibition, Denver, CO, October 6-9, 1996.

Marcondes, F., C. Han, and K. Sepehrnoori: "Implementation of Corner Point Mesh Into a Parallel, Fully Implicit, Equation of State Compositional Reservoir Simulator," 18th International Congress of Mechanical Engineering, Ouro Preto, MG, Brazil, November 6-11, 2005.

Monteagudo, J.E.P., A. Firoozabadi: "Control-Volume Method for Numerical Simulation of Two-Phase Immiscible Flow in Two- and Three-Dimensional Discrete-Fractured Media," *Water Resources Research*, Vol. 40, W07405, April 2004.

Morrow, N.R., I. Chatzis, and H. Lim: "Relative Permeabilities at Reduced Residual Saturation," *J. Can. Pet. Techol.*, 62-69, Jul-August 1985.

Morrow, N.R. and B. Songkran: *Surface Phenomena in Enhanced Oil Recovery*, D. O. Shah (ed.), Plenum Press, New York City, 387-411, 1982.

Naimi-Tajdar, R., C. Han, K. Sepehrnoori, T.J. Arbogast, and A.M. Miller: "A Fully Implicit, Compositional, Parallel Simulator for IOR Processes in Fractured Reservoir," *SPE Journal*, Vol. 12, No. 3, pp. 367-381, September 2007.

Nalla, G.: "Implementation of the chemical flooding module in GPAS," M.S. thesis, The University of Texas at Austin, TX, 2002.

Nelson, R.C. and G.A. Pope: "Phase relationships in chemical flooding," *SPE Journal, Trans. AIME*, 265, October 1978.

Nolen, J.S. and L.S. Patricia: "Reservoir Simulation on Vector Processing Computers," Paper SPE 9644 presented at the SPE Middle East Oil Technical Conference, Manama, Bahrain, March 1981.

Noorishad, J. and M. Mehran: "An Upstream Finite Element Method for Solution of Transient Transport Equation in Fractured Porous Media," *Water Resources Research*, vol.18, No.3, 588-596, 1982.

Oda, M.: "Permeability Tensor for Discontinuous Rock Mass," *Geotechnique* Vol. 35, No. 4, 483-495, 1985.

Odeh, A.S.: "Unsteady-State Behavior of Naturally Fractured Reservoirs," *SPE Journal*, 60-66; Trans., AIME, vol. 234, March 1965.

Parashar, M., J.A. Wheeler, G. Pope, K. Wang, and P. Wang: "A New Generation EOS Compositional Reservoir Simulator: Part II - Framework and Multiprocessing," Paper SPE 37977 presented at the SPE Symposium on Reservoir Simulation, Dallas, TX, June 8-11, 1997.

Peng, D.Y. and D.B. Robinson: "A New Two Constant Equation of State," *Ind. Engng Chem. Fundam.* Vol. 15, 59, 1976.

Perschke, D.R.: "Equation of State Phase Behavior Modeling for Compositional Simulator," Ph.D. dissertation, The University of Texas at Austin, TX, 1988.

Pope, G.A., W. Wu, G. Narayanaswamy, M. Delshad, M.M. Sharma, and P. Wang: "Modeling Relative Permeability Effects in Gas-Condensate Reservoirs," SPE Reservoir Evaluation & Engineering, April 2000.

Prouvost, L., G.A. Pope, and B.A. Rouse: "Microemultion Phase Behavior: A Thermodynamic Modeling of the Phase Partitioning of Amphiphilic Species," *SPE Journal*, 693-703, October 1985.

Pruess, K. and T.N. Narasimhan: "A Practical method for Modeling Fluid and Heat Flow in Fracture Porous Media," *SPE Journal* vol. 2, 14-26, 1985.

Reed, R.L. and R.N. Healy: Some Physico-Chemical Aspects of Microemulsion Flooding: a Review, *Improved oil recovery by surfactant and polymer flooding*, D.O. Shah and R.S. Schechter (eds.), Academic Press, New York, 1977.

Reid, R.C., J.M. Prausnitz and B.E. Poling: *The Properties of Gases and Liquids*, Fourth Edition, McGraw-Hill, Inc., NY, 1987.

Reiss L.H.: *The Reservoir Engineering Aspects of Fractured Formulation*, Gulf Publishing Company, Houston, 1980.

Reiss, L.H.: *The Reservoir Engineering Aspects of Fractured Formations*, Gulf Publishing Company, Houston, TX, 1982.

Rossen R.H.: "Simulation of Naturally Fractured Reservoirs With Semi-Implicit Source Terms," Paper SPE 5737 presented at the Society of Petroleum Engineerings Symposium on Numerical Simulation of Reservoir Performance, Los Angeles, CA, February 19-20, 1976.

Rossen, R.H. and E.I. Shen: "Simulation of Gas/Oil Drainage and Water/Oil Imbibition in Naturally Fractured Reservoirs," Paper SPE 16982 presented at the Society of Petroleum Engineers Annual Technical Conference and Exhibition, Dallas, TX, September 27-30, 1987.

Saidi, A.M.: “Simulation of Naturally Fractured Reservoirs,” Paper SPE 12270 presented at the Society of Petroleum Engineers Symposium on Reservoir Simulation, San Francisco, CA, November 15-18, 1983.

Sandler, S.I.: *Chemical and Engineering Thermodynamics*, Third Edition, John Wiley & Sons, Inc., 1999.

Scott, S.L., R.L. Wainwright, R. Raghavan, and H. Demuth: “Application of Parallel (MIMD) Computers to Reservoir Simulation,” Paper SPE 16020 presented at the 9th SPE Symposium on Reservoir Simulation, San Antonio, TX, 1987.

Shi, S.: “Derivation for Jacobian Matrix in GPAS,” Internal Report at the Center for Petroleum and Geosystems Engineering, the University of Texas at Austin, TX, June 2003.

Smith, J.M. and H.C. Van Ness: *Introduction to Chemical Engineering Thermodynamics*, McGraw-Hill, Inc., NY, 1975.

Snow, D.T.: “Anisotropic Permeability of Fractured Media,” *Water Resources Research*, Vol. 5, No. 6, p. 1273-1289, 1969.

Soave, G.: “Equilibrium Constants From a Modified Redlich-Kwong Equation of State,” *Chem. Engng Sci.* vol. 27, 1197, 1972.

Thomas, L.K., T.N. Dixon, and R.G. Pierson: “Fractured Reservoir Simulation,” *SPE Journal*, 42-54, February 1983.

Thompson, J.F., A.U.A Warsi, and C.W. Mastin: *Numerical Grid Generation-Foundations and Applications*, Elsevier Science Publishing, New York, 1985.

Van Golf-Racht, T.D.: *Fundamentals of Fractured Reservoir Engineering,* Development in Petroleum Science No. 12, Elsevier Scientific Pub. Co., Amesterdam, 1982.

Wallis, J.R., J.A. Foster, R.P. Kendall: “ A New Parallel Iterative Linear Solution Method for Large-scale Reservoir Simulation,” Paper SPE 21209 presented at the SPE Symposium on Reservoir Simulation, Anaheim, California, February 17-20, 1991.

Wang, P., I. Yotov, M. Wheeler, T. Arbogast, C. Dawson, M. Parashar, and K. Sepehrnoori: “A New Generation EOS Compositional Reservoir Simulator: Part I-Formulation and Discretization,” Paper SPE 37979 Proceedings of the SPE Reservoir Simulation Symposium, Dallas, TX, June 1997.

Wang, P., S. Balay, K. Sepehrnoori, J. Wheeler, J. Abate, B. Smith, G.A. Pope: "A Fully Implicit Parallel EOS Compositional Simulator for Large Scale Reservoir Simulation," Paper SPE 51885 presented at the SPE 15th Reservoir Simulation Symposium, Houston, TX, February 14-17, 1999.

Warren, J.E. and P.J. Root: "The Behavior of Naturally Fractured Reservoirs," *SPE Journal*, 245-255, September 1963.

Wheeler, M.F., S. Bryant, J. Eaton, Q. Lu, M. Peszynska, T. Arbogast, and I. Yotov: "A Parallel Multiblock/Multidomain Approach for Reservoir Simulation," Paper SPE 51884 presented at the 15th SPE Reservoir Simulation Symposium, Houston, TX, February 1999.

White C.D. and R.N. Horne: "Computing Absolute Transmissibility in the presence of Fine-Scale heterogeneity," Paper SPE 16011 presented at the Society of Petroleum Engineers Symposium on Reservoir Simulation, San Antonio, TX, February 1-4, 1987.

Winsor, P.A.: *Solvent Properties of Amphiphilic Compounds*, Butterworths, London, 1954.

Wu, Y.S. and K. Pruess: "A Multiple-Porosity Method for Simulation of Naturally Fractured Reservoirs," Paper SPE 15129 presented at the Society of Petroleum Engineers California Regional Meeting, Oakland, CA, Aprril 2-4, 1986.

Vita

Farhad Tarahhom was born on September 21, 1974, the son of Yadollah Tarahhom and Cobra Bahmei, in Bandarmahshahr, Iran. He joined the National Iranian Oil Company (NIOC) in September 1990 and worked until March 1997. Meanwhile he entered the Azad University, Omidieh Branch in 1992 and received his B.Sc. degree in Chemical Engineering (Utilization of Petroleum Resources) on February 1997. On September 1997, he entered The University of Tehran and received a Master of Science in Chemical Engineering (Hydrocarbon Reservoir Engineering) in January 2000. After three years working as a petroleum reservoir engineer with Tehran Energy Consultant Company, he entered the Graduate School of The University of Texas at Austin in Summer 2001. For his dissertation, he developed an implicit full-tensor compositional dual porosity simulator for naturally fractured reservoirs.

Farhad is married to Ghazal Dashti.

Permanent Address: 3561 Lake Austin Blvd, Apt# D
Austin, TX 78703, USA

This dissertation was typeset with LaTeX 2_ε[1] by the author.

---

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written

---

by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay, James A. Bednar, and Ayman El-Khashab.